Universidad de Valladolid

Grado en Estadística

Construcción de un modelo de goles esperados para los partidos de la Copa Mundial de la FIFA del año 2018

Trabajo de Fin de Grado

Autor

Jorge San José Lorza

Tutora

María Teresa González Arteaga

Resumen

Durante los últimos años la Estadística avanzada se ha abierto paso en el mundo del fútbol hasta tal punto que la mayoría de clubs cuentan en su organigrama con un departamento encargado de realizar análisis económicos y deportivos basados en Big Data.

En este trabajo se realiza una aproximación a la Estadística en el ámbito del fútbol por medio de una de las métricas que más popularidad ha adquirido en los últimos años, los *expected goals (xG)* o goles esperados. Esta métrica mide el número de goles que un jugador o equipo debería haber conseguido dadas las ocasiones de las que dispuso, permitiendo realizar comparaciones de rendimiento ofensivo tanto individuales como colectivas.

En el presente documento se expone el proceso de construcción de un modelo de goles esperados para los partidos disputados en la Copa Mundial de la FIFA en el año 2018. Se han empleado los datos de los partidos de las cinco grandes ligas europeas en la temporada 2017-2018 para generar el modelo, ya que como se explica más adelante, estas competiciones suponen un espacio muestral representativo de los partidos del Mundial.

Como parte del proceso de creación del modelo se detallan las variables consideradas de interés, así como los procedimientos seguidos para su obtención. Desde el punto de vista teórico, se ha utilizado el algoritmo *Gradient Boosting*, y en concreto su implementación *XGBoost*, para la construcción del modelo, ya que es una de las técnicas de clasificación más utilizadas en la actualidad debido a sus buenos resultados y eficiencia.

Abstract

Over the last few years, advanced Statistics has made its way into the world of football to such an extent that most clubs have a department in charge of economic and sporting analysis based on Big Data.

In this paper, an approach to Statistics related to soccer is developed by means of one of the most popular metrics in recent years, the expected goals (xG). This metric measures the number of goals a player or team should have scored given the chances they had, allowing comparisons of both individual and collective offensive performance.

This dissertation outlines the process of building a model of expected goals for the matches played in the 2018 FIFA World Cup. Data from matches of the five major European leagues in the 2017-2018 season have been used to generate the model, since, as it is explained below, these competitions suppose a representative sample space of World Cup matches.

As part of the model creation process, the variables considered significant are detailed, as well as the procedures followed to obtain them. From a theoretical point of view, the Gradient Boosting algorithm, and specifically its implementation XGBoost, has been used to build the model, as it is one of the most widely used classification techniques due to its good results and efficiency.

Agradecimientos

En primer lugar, a todos los profesores involucrados en mi formación durante estos cinco años, y en especial, a mi tutora, Teresa, no solo por su apoyo en este trabajo fin de grado, sino por darme la oportunidad de realizar el mismo sobre una temática de gran interés para mi.

Gracias a mis compañeros, por su ayuda siempre que lo he necesitado y por convertir estos años en una etapa inolvidable de la que me llevo grandes amistades.

A mi pareja, Marta, y mis amigos, por confiar en mí y animarme en los momentos más difíciles, siendo una parte fundamental para que este trabajo saliese adelante.

Y sobre todo, gracias a mi familia, por su esfuerzo durante estos años en los que siempre me han brindado todas las facilidades posibles para que mi única preocupación fuese completar mi formación académica.

Índice

Índice de figuras

Índice de cuadros

1. Introducción

Durante las últimas décadas, el mundo de la tecnología ha sufrido grandes avances que han desembocado en una sociedad altamente tecnológica y digitalizada. Como consecuencia, a diario se producen ingentes cantidades de datos, hasta tal punto que se estima que durante el año 2021 cada persona producirá, de media, 3.4MB de datos por segundo [1].

Estos grandes volúmenes de datos, tras un correcto proceso de depuración y tratamiento, pueden ser utilizados para extraer información muy valiosa, tanto desde el punto de vista económico como social. A este proceso de almacenamiento, depuración y tratamiento, se le conoce bajo el término de Big Data.

A día de hoy, todas las grandes empresas cuentan con profesionales en Big Data, que trabajan en busca de información que les ayude a conseguir sus objetivos de manera más económica y segura. Una de las actividades empresariales que más dinero mueve en el mundo, es el deporte, y en concreto, el fútbol. Se estima que en España, durante el año 2021, la actividad económica vinculada al fútbol de élite supuso el 1.37 % del PIB y más de 185.000 empleos [2]. Como no podría ser de otra forma, el Big Data se ha convertido en un pilar básico dentro de las entidades deportivas.

En los clubs de fútbol, el uso del Big Data no se restringe únicamente al aspecto financiero o social, sino que también es de gran utilidad en los análisis deportivos, tanto de jugadores como de entrenadores y equipos. Una de las métricas que se usan para este fin y que en la actualidad ha conseguido una gran popularidad, sobre todo entre los aficionado y la prensa deportiva, es la de **goles esperados**, o *expected goals (xG)*. Mediante esta métrica se establece una **valoración** de la claridad **de las ocasiones de gol** de las que dispuso un jugador, o equipo, a lo largo de uno o más partidos.

La **interpretación** de los xG es bastante sencilla. Si un jugador tiene durante un partido un xG acumulado de 1.27, esto significa que ha dispuesto de ocasiones suficientes como para conseguir marcar 1.27 goles. Evidentemente, es imposible marcar un número no exacto de goles, pero este valor decimal se produce por cómo se calcula la métrica. El valor de goles esperados, para un jugador en un partido, es la suma de las probabilidades de finalizar en gol, otorgadas por el modelo, a cada tiro que realizó. Por ello, si el jugador ha efectuado cuatro lanzamientos, con probabilidades de éxito de 0.30, 0.42, 0.27 y 0.28, su xG será de 1.27.

La creación de un modelo de goles esperados se reduce a generar un **clasificador binario** en el que los posibles valores de la respuesta son: "Gol" y "No-Gol". Para llevar a cabo la clasificación, se pueden tener en cuenta múltiples variables, como pueden ser: la distancia y el ángulo respecto a la portería desde el lugar de lanzamiento, el pie con el que se realiza el tiro... El modelo más sencillo, para llevar a cabo el clasificador, es la **regresión logística**. Mediante ella, no solo se puede obtener una clasificación de las instancias a valorar, sino que, además, se obtiene la **probabilidad** de pertenencia de la instancia a la clase de interés, en nuestro caso "Gol". Esta probabilidad es el valor que realmente se utiliza como métrica.

En el presente Trabajo Final de Grado, en adelante TFG, se presenta la creación de un modelo de goles esperados para su uso con los partidos de la Copa Mundial de la FIFA disputada en Rusia en el año 2018. Como parte del proceso de creación del modelo, se pueden distinguir las siguientes fases:

- *Depuración del conjunto de datos a utilizar*: No existen datasets creados explícitamente para este fin, por lo que se va a tener que extraer la información requerida de un conjunto de datos de gran tamaño.
- *Generación de las variables a incluir en el modelo*: A partir del dataset, ya depurado, es necesario decidir qué variables creemos importantes para el modelo, así como definir los procedimientos necesarios para construirlas a partir de los datos contenidos en el dataset.
- *Generación del modelo*: Una vez definidas las variables de interés, se debe construir un modelo de clasificación que evalúe la probabilidad de que una ocasión, o tiro, finalice en gol. En este TFG se expone una solución haciendo uso de la implementación XGBoost del algoritmo Gradient Boosting.

Los datos utilizados en este TFG [3] fueron recolectados por la empresa *WyScout*, y contienen información sobre los partidos disputados, durante la temporada 2017-2018, en las cinco grandes ligas europeas, que son las primeras divisiones de *Alemania*, *España*, *Francia*, *Inglaterra* e *Italia*. Los datos de estas competiciones, son los empleados en la fase de construcción del modelo. Evidentemente, en este dataset también se contiene la información de los partidos para los que se construye el modelo: la Copa Mundial de la FIFA del año 2018.

El objetivo principal de este TFG es la construcción de un modelo de goles esperados para los partidos del Mundial de Rusia del 2018. Como pasos previos a alcanzar este objetivo, se establecen las fases y objetivos descritos anteriormente, que guían el proceso del proyecto.

Para una correcta exposición del trabajo realizado, el presente TFG se estructura de la siguiente forma:

- **Capítulo 2. Marco Teórico.** Se explican los conceptos teóricos sobre los que se sustenta la solución propuesta para el modelo de goles esperados. Se introducen los ensembles de clasificadores y el algoritmo de Gradient Boosting, así como su implementación XGBoost empleada en este TFG.
- **Capítulo 3. Conjunto de datos proporcionado por WyScout.** Se realiza una exposición sobre el contenido del conjunto de datos utilizado para este trabajo, así como los procesamientos necesarios para su adecuación al problema.
- **Capítulo 4. Contextualización.** En primer lugar, se realiza una exposición sobre la métrica de goles esperados y su utilidad y relevancia. En segundo lugar, se exponen los motivos por los que se considera que las cinco grandes ligas europeas suponen un buen espacio muestral para el problema.

- **Capítulo 5. Conjunto de datos para el modelo.** Se detallan las condiciones que debe reunir una ocasión para ser valorada por nuestro modelo, y las variables que se incluirán en el mismo.
- **Capítulo 6. Búsqueda del modelo XGBoost y resultados.** Se define el proceso seguido para encontrar el modelo XGBoost que mejor se ajusta al problema del cálculo de los goles esperados y se presentan los resultados obtenidos.
- **Capítulo 7. Conclusiones.** Se realiza una valoración sobre el aprendizaje que ha supuesto este TFG y se establecen las líneas futuras de trabajo.

2. Marco teórico

En este capítulo se van a exponer los fundamentos teóricos sobre los que subyace el modelo de goles esperados propuesto en el presente trabajo. Se ha de recordar, que la generación de un modelo de goles esperados se reduce a la creación de un clasificador binario, que determine si la instancia debe considerarse como "Gol" o como "No-Gol". Para resolver este problema, se ha optado por utilizar *XGBoost*, que es una implementación y generalización del *Gradient Boosting*. Con el fin de comprender su funcionamiento, se comenzará hablando de los árboles de decisión, que son los clasificadores en los que se basa este algoritmo. A continuación, se definirá el concepto de *ensemble de clasificadores*, así como los dos tipos principales existentes. Por último, se realizará una exposición sobre el Gradient Boosting y las ventajas del XGBoost.

2.1. Árboles de decisión

El origen de los árboles de decisión se remonta al año 1963, con la publicación del artículo *Problems in the Analysis of Survey Data, and a Proposal* por parte de *James Nelson Morgan* y *John A. Sonquist* [4]. En la actualidad, se trata de uno de los algoritmos de aprendizaje supervisado más utilizados.

Los árboles de decisión se basan en un enfoque descendiente que tiene como objetivo generar un modelo de predicción que divida el espacio de los predictores en agrupaciones de observaciones con valores similares para la variable respuesta o dependendiente. Para dividir el espacio muestral en sub-regiones se aplican una serie de reglas de decisión, con el objetivo de que cada sub-región contenga la mayor proporción posible de individuos de una de las poblaciones. En el caso de que una sub-región siga conteniendo datos de diferentes clases, se divide en nuevas regiones más pequeñas hasta que estas integren datos de la misma clase. Una vez generado el árbol se pueden usar diversas métricas para evaluar su eficiencia siendo, en el caso de la clasificación, el *índice de impureza de Gini* y la *entropía* las más utilizadas.

2.1.1. Estructura general

La división descendente en función de las reglas de decisión proporciona a este algoritmo una representación visual similar a la de un árbol, tal y como se puede observar en la Figura 1

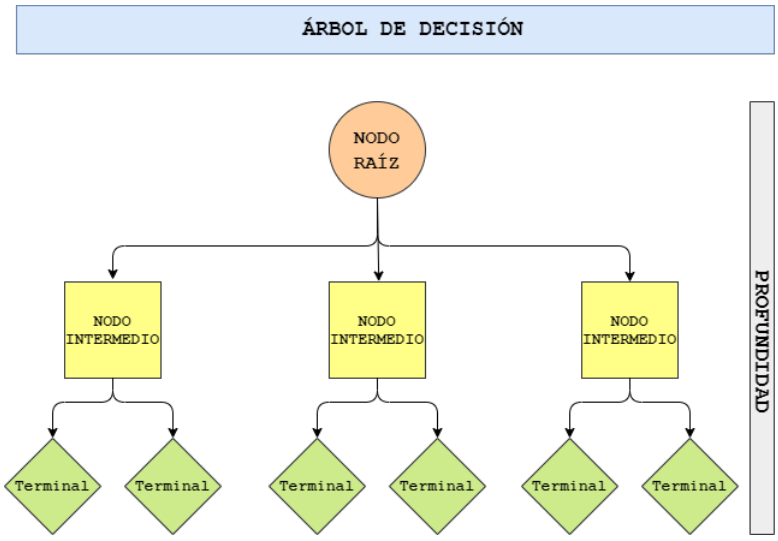

Figura 1: Estructura genérica de un árbol de decisión.

Dentro de la jerarquía del árbol existen tres tipos de nodos diferentes:

- *Nodo raíz*: es el nodo del que surgen todos los demás, por ello está asociada siempre con la variable más importante del modelo ya que este nodo significa la primera ramificación del árbol. Emplear una variable poco significativa en este nodo no sería de utilidad, ya que llevaría a un desaprovechamiento de la información disponible.
- *Nodos intermedios*: representan las reglas de decisión necesarias para llevar a cabo la clasificación.
- *Nodos terminales*: como su propio nombre indica son los últimos nodos del esquema. Representan la clase definitiva a la que pertenece un individuo.

Aunque en la Figura 1 solo se muestre un nodo intermedio en cada ramificación es habitual que existan nodos intermedios que cuelguen de otros hasta que finalmente se produzcan los nodos terminales. Además, las ramas no tienen porque tener el mismo número de nodos intermedios ni terminales, como ocurre en la Figura 1, sino que la estructura del árbol estará determinada por el conjunto de datos a modelar por lo que no tiene por qué presentar una estructura simétrica.

2.1.2. Profundidad y poda

Se define el *camino* a un nodo B desde un nodo A como el conjunto ordenado de nodos por los que se ha de pasar para alcanzar B partiendo desde A. Debida a la naturaleza estructural de este algoritmo está garantizado que dichos recorridos son únicos, por lo que un nodo no puede ser alcanzado por

diversos caminos.

Se denomina *profundidad de un nodo* al número de nodos que conforman el camino al respectivo nodo desde el nodo raíz. Por ejemplo, para el caso de un nodo terminal que pende de un nodo intermedio que a su vez cuelga de otro nodo intermedio dependiente del nodo raíz se tiene que su profundidad es 3, ya que se han atravesado tres nodos hasta llegar al nodo buscado. De manera similar se define la *profundidad del árbol* como la longitud del camino más largo que se puede hacer desde el nodo raíz a cualquier otro nodo del árbol.

La profundidad del árbol es un parámetro crítico en los árboles de decisión. Teniendo en cuenta que cada nodo representa una disyunción sobre alguna de las variables del espacio muestral, una profundidad demasiado elevada puede conducir a un sobreajuste ya que es posible que muchos de esos nodos no lleven a una mejor generalización de la clasificación sino que únicamente sirvan para mejorar la precisión sobre el conjunto de datos con el que se esté trabajando.
Para limitar la profundidad de un árbol existen dos enfoques distintos:

- *Limitar la profundidad de antemano*: bajo esta idea el árbol puede crecer con normalidad hasta alcanzar una profundidad máxima establecida en la que los posibles nodos intermedios se convierten en nodos terminales atendiendo a la distribución de los ejemplos que contiene.
- *Podar el árbol una vez generado*: el árbol se genera sin ningún tipo de restricción y una vez definida su estructura final se produce la poda. Todos aquellos nodos, ya sean terminales o intermedios, que tienen un nivel de profundidad superior al fijado son eliminados del árbol, convirtiéndose en nodos terminales aquellos nodos intermedios que tienen un nivel de profundidad igual al máximo establecido.

Pero no solo debemos tener en cuenta la profundidad de un árbol para intentar paliar el sobreajuste o mejorar la precisión de la clasificación, sino que existen otros parámetros que se deben fijar antes de comenzar a generar el árbol y que pueden determinar el éxito del mismo. Algunos de esos parámetros son el número mínimo de observaciones para dividir un nodo o considerarlo terminal, el número máximo de nodos terminales que se van a poder dar en el árbol o el número máximo de atributos a considerar para realizar un ramificación.

2.1.3. Ventajas y limitaciones

El principal punto de fuerte de este algoritmo es su alto grado de interpretación comparado con otros modelos de Aprendizaje Automático. Su estructura en árbol permite conocer perfectamente en base a que criterios se realiza la clasificación lo que además facilita la comprensión de las conclusiones obtenidas. Siguiendo con la comparación con otros algoritmos de Aprendizaje Automático, los árboles de decisión destacan por no precisar de un gran conjunto de datos de entrenamiento siendo incluso

tolerante con datos incompletos (ausencia de valores en algunos campos). Además, permite utilizar distintos tipos de datos en un mismo conjunto.

Por el contrario, uno de sus principales inconvenientes es que tiende al sobreajuste a pesar de que hemos visto que este fenómeno se puede tratar de prevenir fijando una profundidad máxima de antemano o realizando una poda a posteriori. Otro de los principales problemas relacionados con este algoritmo es que no es capaz de realizar grandes predicciones por si solo, siendo además bastante sensible a cambios en el conjunto de datos. Para obtener buenos resultados precisa de ser utilizado en conjunto con otros árboles de decisión que doten de un mayor nivel de precisión en la clasificación y de una mayor robustez. Al no dar buenos resultados por si solo y precisar de ser empleado en conjunto con otros clasificadores de su misma naturaleza, se considera que un árbol de decisión es un *clasificador débil*. Cuando varios clasificadores débiles se utilizan para generar un único clasificador más preciso y robusto se dice que conforman un *ensemble*.

2.2. Ensembles de clasificadores

Los ensembles [5] se pueden categorizar en *homogéneos* si todos los clasificadores débiles son del mismo tipo o *heterogéneos* si se usan clasificadores de distinta naturaleza. En este trabajo vamos a centrarnos en los homogéneos, siendo los árboles de decisión los clasificadores débiles empleados.

Dentro del enfoque homogéneo existen dos técnicas principales a la hora de construir el ensemble: *Bagging* y *Boosting*. A continuación se explicarán las principales diferencias entre ambos, así como las ventajas y desventajas del uso de uno u otro [5].

2.2.1. Bagging

Su origen se remonta a mediados de los años 90 [6], y a pesar de no ser el primer método de ensemble conocido si que supuso un impulso definitivo del campo. Su nombre proviene de la idea de *Bootstrap AGGregatING* en la que se sustenta.

El método ideal de Bagging consiste en obtener diversos conjuntos de datos de tamaño n, independientes, de manera aleatoria y emplearlos para construir clasificadores débiles. Cada uno de estos clasificadores, o hipótesis, genera su propia predicción y la combinación de todas ellas da lugar a la clasificación realizada por el ensemble (ver Figura 2). Para combinar las predicciones se emplea el *voto por mayoría*, siendo la clasificación del ensemble la predicción que más se haya repetido entre todas las hipótesis. En la práctica, no es frecuente contar con diversos conjuntos de datos independientes sino que se dispone de un único conjunto de datos etiquetados. Para simular dichos conjuntos, se realiza un *muestreo con reemplazamiento* sobre el conjunto inicial. Estos conjuntos no son del todo independientes, como se planteaba en el escenario ideal, pero permiten alcanzar grandes resultados.

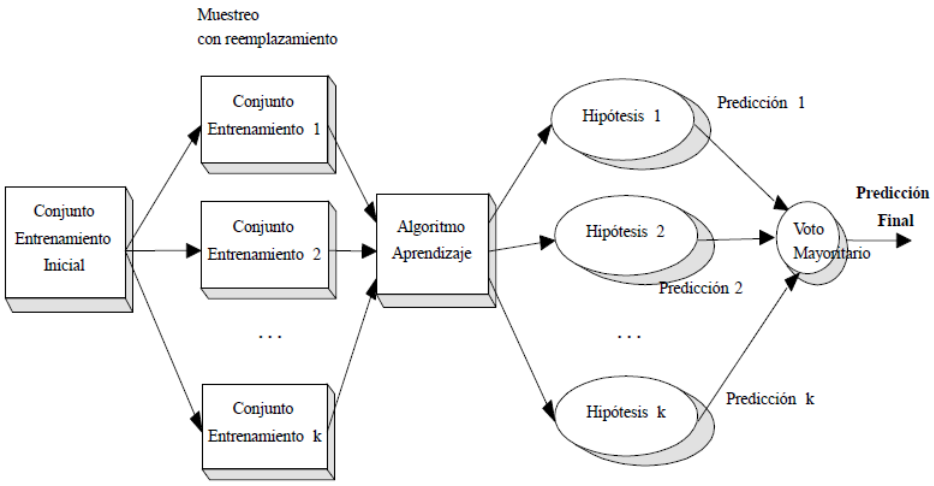

Figura 2: Esquema general de la estructura de un ensemble bagging.

En la generación de un ensemble de este tipo, por tanto, existen dos pasos fundamentales: la *generación de los modelos* y la *clasificación*.

Para generar los diversos modelos, o clasificadores débiles, los pasos que debemos seguir son:

1. Localizar un conjunto de entrenamiento, que denominaremos D.
2. Elegir el tamaño del ensemble, L, y el clasificador base, en nuestro caso árboles de decisión.
3. Fijar un número de instancias, n, en cada subconjunto de entrenamiento que generaremos.
4. Para i = 1, ..., L hacer:
   - Generar D_i obteniendo n muestras con reemplazamiento de D.
   - Entrenar el clasificador base C_i con D_i.
   - Añadir el clasificador C_i al ensemble E.

Una vez tenemos constituido el ensemble, podemos proceder a realizar la clasificación. Supongamos que queremos clasificar una instancia que denotaremos x. Para cada modelo C_i del ensemble obtenemos el voto dato a cada clase w_k: $v_{i,k}$. Este valor será 1 si C_i predice w_k y 0 en caso contrario. Después, para cada clase w_k se calcula V_k como el sumatorio de todos los votos dados a dicha clase a lo largo de todos los clasificadores. Finalmente, se realiza la clasificación en la clase w_i cuyo valor V_i se máximo.

Bagging supone un método de clasificación altamente atractivo por su simplicidad y potencial entrenamiento paralelo. Pero realmente no hay una respuesta sencilla sobre por qué funciona. Si las salidas de los clasificadores fueran independientes y el error verdadero fuese inferior a 0.5, el voto mayoritario mejora la tasa de error. Esto ocurriría en el escenario ideal, ya que en la práctica los clasificadores no son independientes ya que se han entrenado con conjuntos obtenidos mediante muestreo con reemplazamiento. De forma intuitiva se atribuye su eficacia a a la utilidad de combinar clasificadores diversos, mientras que de forma teórica se justifica mediante la descomposición *Sesgo-Varianza*.

La descomposición Sesgo-Varianza trata de explicar el error de una hipótesis desde un análisis teórico basado en el origen de la misma. El término sesgo hace referencia al error persistente que comete el algoritmo de aprendizaje sobre el problema, mientras que la varianza esta asociada a la elección particular del conjunto de entrenamiento. La suma de ambos conceptos da lugar al error total del clasificador, siendo frecuente que los modelos más sencillos cuenten un sesgo elevado y una varianza baja mientras que en los modelos más complejos sucede justo lo contrario. Hay evidencia teórica y experimental de que Bagging funciona porque reduce el componente de la varianza e incluso en algunos ejemplos concretos también reduce el sesgo siempre y cuando sea elevado.

Algunos de los algoritmos más conocidos que se basan en Bagging son *Random Forest* y *Random Subspace*.

2.2.2. Boosting

En los algoritmos de Boosting los clasificadores base no son generados en paralelo como ocurre en el Bagging, sino que se trata de un proceso iterativo en el que los nuevos modelos generados están influenciados por el comportamiento de los anteriores. Gracias a este planteamiento es posible forzar al algoritmo a que se centre en los ejemplos que han sido mal clasificados anteriormente y mejorar de esta manera la eficiencia del modelo.

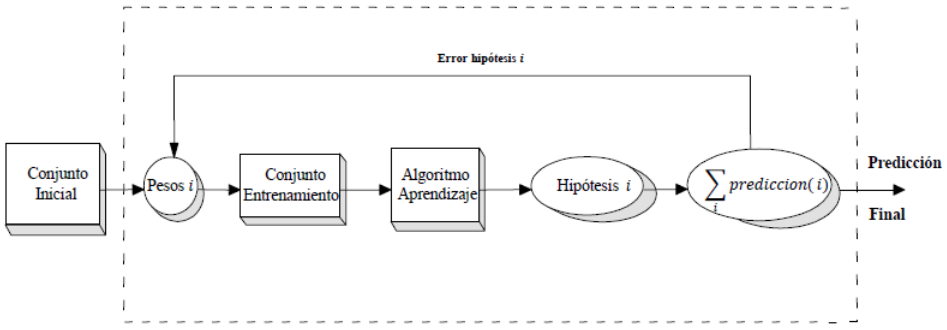

Figura 3: Esquema general de la estructura de un ensemble boosting.

Como podemos ver en la Figura 3 en un ensemble Boosting existe un lazo de retroalimentación de los pesos en función del error de la última predicción realizada. Mediante esta retropropagación del error se busca dar un mayor peso a aquellos ejemplos mal clasificados forzando a los clasificadores débiles a centrarse en ellos con un mayor énfasis. Esto hace que cada hipótesis tenga un peso distinto en función de su calidad, mientras que en el Bagging todas tenían el mismo peso. Debido a este hecho, en Boosting se emplea el *voto ponderado* en lugar del voto por mayoría.

El proceso de creación de un ensemble de este tipo se puede resumir en los siguientes pasos [7]:

1. Sean $\hat{f}(x) = 0$ nuestro predictor lineal y los residuos igual a las observaciones, $r_i = y_i$ en un primer paso.
2. Sea B el número de árboles que constituyen el ensemble. Repetir para cada uno de los árboles a generar (b = 1, ..., B):
   - Ajustar un árbol $\hat{f}^b$ de tamaño *d* con *(X,r)*. Siendo X las variables y r los residuos.
   - Actualizar $\hat{f}(x)$ haciendo uso del parámetro *(shrinkage)* o de decrecimiento (λ):
   $$\hat{f}(x) = \hat{f}(x) + \lambda \hat{f}^b$$
   - Actualizar los residuos: $r_i = r_i - \lambda \hat{f}^b$
3. Obtenemos el clasificador final: $\hat{f}(x)$

En vista del proceso de conformación del ensemble, podemos destacar tres hiperparámetros:

1. *El número de árboles*, *B*. Un valor de B demasiado elevado puede acabar provocando sobreajuste. Se suele determinar mediante validación cruzada.
2. *El parámetro de decrecimiento*, λ. Controla la velocidad a la que el ensemble aprende. Los valores típicos son 0.01 y 0.001, aunque el mejor valor depende del problema. Un valor demasiado pequeño de λ puede requerir del uso de un valor elevado para B si se quieren conseguir buenos resultados.
3. *El número de divisiones* o tamaño del árbol, *d*. Controla la complejidad del modelo. A menudo d = 1 proporciona buenos resultados, tratándose de un modelo aditivo.

En la actualidad los ensembles de este tipo son altamente populares debido al gran rendimiento que ofrecen. Experimentalmente se ha demostrado que es capaz de reducir el sesgo en sus primeras iteraciones y la varianza en las últimas, pero no se ha encontrado una justificación teórica para este hecho. Si se quiere obtener una explicación de carácter teórico sobre la eficacia del Boosting se debe recurrir a la *teoría del margen*.

El *margen* es una medida de la confianza del clasificador en sus predicciones. De forma intuitiva se puede definir como la diferencia entre la fracciones de votos correctos e incorrectos. Si el margen es positivo entonces la clasificación es correcta, y en caso contrario es incorrecta. De manera formal se define como:

$$m(x_i) = \frac{y_i \sum_t \alpha_t C_t(x_i)}{\sum_t |\alpha_t|}$$

donde:

- x_i es el ejemplo a clasificar.
- y_i clase a la que pertenece realmente x_i.
- $C_t(x_i)$ la clase que asigna a x_i el clasificador t-ésimo.
- α_t el peso del clasificador.

En la bibliografía [5] se indica que experimentalmente se ha podido comprobar que añadir clasificadores a un ensemble con un error de resubstitución nulo, disminuye el error esperado en la fase de validación o test, algo que es totalmente anti intuitivo (Navaja de Occan).

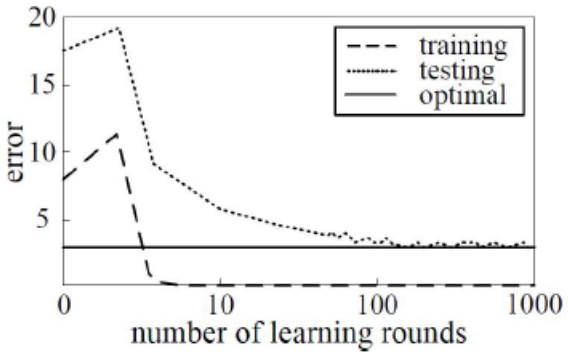

Figura 4: Simulación de la evolución de los errores en un ensemble boosting [5].

Este hecho se puede explicar en términos de la teoría del margen. Se sabe que los clasificadores con mayor margen tienen un menor error verdadero (mayor margen de la instancia con mínimo margen), por lo que se puede considerar al margen como una cota superior teórica del error verdadero [5].

Existen múltiples variantes de Boosting, como pueden ser los algoritmos de la familia *AdaBoost* que fueron pensados inicialmente para clasificadores binarios, *Local boost* que considera el error sobre cada instancia en vez del error medio, *Robust boost* que minimiza el número de ejemplos cuyo margen es menor que un cierto umbral, *LogitBoost* que propone un ensemble de regresores logísticos... Pero

en este trabajo nos centraremos en el *Gradient Boosting* y su implementación *XGBoost*, ya que es una de las opciones más utilizadas en la actualidad por los buenos resultados que ofrece.

2.3. Gradient Boosting

Gradient Boosting es un algoritmo de optimización que utiliza la técnica del descenso de gradiente sobre una función de perdida diferenciable para calcular los pesos de los clasificadores añadidos al ensemble.

2.3.1. ¿Qué es el descenso de gradiente?

El método del descenso de gradiente [8] es un algoritmo de optimización que permite converger hacia el valor mínimo de una función mediante un proceso iterativo. En el caso del Aprendizaje Automático se emplea para minimizar una función que mide el error de predicción del modelo en el conjunto de datos.

Para identificar el mínimo de la función el método del descenso de gradiente propone el cálculo de la derivada parcial respecto a cada parámetro del modelo en el punto que se quiera evaluar. Mediante esta derivada se obtiene tanto el valor como el sentido en el que se encuentra el mínimo más cercano. El resultado de dicha derivada es restado a cada uno de los parámetros multiplicado por la velocidad de aprendizaje (α), que indica lo rápido que converge el algoritmo.

Denominando como J a la función a minimizar y como θ a los parámetros del modelo, el algoritmo básico [9] para el descenso de gradiente es:

1. Establecer unos valores iniciales para θ.
2. Fijar la velocidad de aprendizaje del algoritmo, α.
3. Hallar la derivada de J en el punto θ.
4. Actualizar el valor de θ con la expresión: $\hat{\theta} = \theta - \alpha \nabla J(\theta)$
5. Comprobar si el cambio producido en el valor de θ es inferior a uno fijado previamente, conocido como *criterio de parada*.
6. En caso afirmativo finaliza la ejecución. En caso contrario, volver al paso 3.

Gradient Boosting se basa en esta idea para minimizar una función $\hat{\mathrm{F}}(\mathrm{x})$ que proporciona la mejor aproximación a la variable respuesta correspondiente.

2.3.2. Formulación del algoritmo de Gradient Boosting

El algoritmo de Gradient Boosting, como cualquier otro algoritmo de clasificación, tiene como objetivo generar un modelo que se ajuste lo más posible a la realidad. Para ello, es necesario introducir un función de pérdida que mida el grado de similitud entre los valores predichos y los reales [10]. Esta función se denota como *L(y,F(x))* y el objetivo es encontrar la función *F(x)* que la minimice como podemos observar en la Ecuación 1.

$$\hat{F} = \underset{F}{\operatorname{argmin}}\ E_{x,y}[L(y, F(x)) \tag{1}$$

Esta función $\hat{F}$ se puede reformular [11] en términos de los clasificadores base que conforman el ensemble (Ecuación 2).

$$\hat{F}(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const. \tag{2}$$

donde:

- El ensemble se considera formado por M clasificadores débiles.
- Cada clasificador débil, i, es representado por la función $h_i(x)$.
- El peso asociado al clasificador i-ésimo se denota como γ_i.
- El valor *const* está asociado al problema y no depende de los clasificadores débiles que componen el ensemble.

Considerándose el principio de minimización del riesgo empírico, el problema se reduce a encontrar una aproximación a $\hat{F}(x)$ que minimice el valor medio de la función de pérdida (el riesgo empírico). Computacionalmente este enfoque es inviable ya que requiere del cálculo de la mejor función h para una función de pérdida cualquiera L. Por ello, se opta por un enfoque basado en el descenso de gradiente [11].

Sea $\{\{x_i, y_i\}\}_{i=1}^{n}$ el conjunto de entrenamiento, *L(y,F(x))* la función de pérdida y M el número de clasificadores débiles, el algoritmo se puede resumir en los siguientes pasos:

1. Inicializar el modelo con un valor constante para la función F:

$$F_0(x) = \underset{\lambda}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3}$$

2. Para m = 1, ..., M, hacer:

   a) Para i = 1, ..., n, calcular los respectivos pseudo-residuos:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{4}$$

   b) Ajustar un clasificador débil, $h_m(x)$ a estos pseudo-residuos, es decir, considerando $\{\{x_i, r_{im}\}\}_{i=1}^{n}$.

   c) Calcular λ_m como:

$$\lambda_m = \underset{\lambda}{\operatorname{argmin}} \sum_{i}^{n} L(y_i, F_{m-1}(x_i) + \lambda h_m(x_i)) \tag{5}$$

   d) Actualizar el modelo:

$$F_m(x) = F_{m-1}(x) + \lambda_m h_m(x) \tag{6}$$

3. Obtenemos la solución como $F_m(x)$.

En el caso de que los clasificadores base sean árboles de decisión, como en este trabajo, se pueden realizar una serie de modificaciones sobre el algoritmo propuesto con el objetivo de mejorar los resultados.

2.3.3. Gradient Boosting y el uso de árboles de decisión

El algoritmo de Gradient Boosting es utilizado frecuentemente con árboles de decisión como clasificadores débiles. Para este caso, *Friedman* propuso una serie de modificaciones al algoritmo de Gradient Boosting con el objetivo de mejorar la calidad de los ajustes de cada uno de los clasificadores débiles [11].

El algoritmo de Gradient Boosting genérico en su m-ésimo paso ajusta el árbol de decisión $h_m(x)$ a los pseudo-residuos correspondientes. Este árbol cuenta con un número de hojas J_m), conteniendo cada una de ellas una región disjunta de las demás. A estas regiones las denotamos como $R_{1m}, ..., R_{J_m m}$ y predicen un valor constante en cada una de ellas. Teniendo en cuenta estas subregiones se puede redefinir el árbol h_m como en la Ecuación 7

$$h_m(x) = \sum_{j=1}^{J_m} b_{jm} 1_{R_{jm}}(x) \tag{7}$$

donde b_{jm} es el valor predicho en la región R_{jm}.

Friedman propuso asignar un peso diferente (λ_{jm}) a cada una de las regiones del árbol m-ésimo en lugar de un peso único (λ_m) para todo el árbol. El impacto de este cambio en el algoritmo de Gradient Boosting se refleja en la Ecuación 6 que es sustituida por la Ecuación 8 y en la Ecuación 5 que es cambiada por la Ecuación 9.

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \lambda_{jm} 1_{R_{jm}}(x) \tag{8}$$

$$\lambda_{jm} = \underset{\lambda}{\operatorname{argmin}} \sum_{x_i R_{jm}} L(y_i, F_{m-1}(x_i) + \lambda) \tag{9}$$

Una de las versiones de Gradient Boosting que más reconocimiento ha ganado durante los últimos años en la comunidad científica es el XGBoost, que se comenta en la siguiente sección, ya que es la versión empleada en este trabajo.

2.4. XGBoost

XGBoost, o *eXtreme Gradient Boosting*, es una implementación y generalización del Gradient Boosting. Fue propuesto en el año 2016 por *Tianqi Chen* [12], quien indicó que *"el nombre de xgboost hace referencia al objetivo de la ingeniería de llevar al límite la explotación de los recursos computacionales existentes para los algoritmos de boosting en árboles de decisión"*.

Las principales ventajas [13] que ofrece el uso de XGBoost en lugar del Gradient Boosting son:

- **Regularización**: permite incluir regularización a la función objetivo, tanto *Ridge (L1)* como *Lasso (L2)*, para tratar de evitar un posible sobre ajuste a través de la selección de variables.
- **Alto nivel de flexibilidad**: el usuario puede definir sus propias funciones objetivo o de pérdida, lo que abre una nueva dimensión a los modelos ya que no hay límites sobre lo que se puede hacer o no.
- **Computación en paralelo**: a pesar de que el Gradient Boosting se basa en una construcción iterativa, XGBoost es capaz de trabajar en paralelo. En cada nodo no terminal se establecen caminos independientes unos de otros sobre los que XGBoost realiza cómputos en paralelo.
- **Empleo de datos incompletos**: la gestión de los datos perdidos o *missings* es muy eficiente, ya que es capaz de aprender qué camino tomar para estos valores gracias a que en la fase de generación del modelo experimenta múltiples opciones con ellos.
- **Poda**: una vez generados los árboles del tamaño indicado procede a realizar una poda sobre las hojas que tienen una pérdida negativa, lo que hace que la estructura de los árboles sea más flexible. En su lugar Gradient Boosting propone no continuar la construcción del árbol una vez se produce la primera pérdida negativa.

3. Conjunto de datos proporcionado por WyScout

A lo largo de este capítulo se va a realizar una exposición sobre el conjunto de datos empleado para llevar a cabo el presente trabajo. En primer lugar, se introduce su fuente de procedencia, WyScout. A continuación, se describe el contenido del conjunto, así como su organización. Por último, se detallan los tratamientos a los que han sido sometidos antes de su utilización.

3.1. ¿Qué es WyScout?

WyScout es la empresa líder en la recolección de datos en el mundo del fútbol. En su base de datos cuentan con los vídeos de los partidos de más de 850 competiciones, repartidas por más de 90 países y que se actualizan semanalmente con unos 1800 partidos nuevos.

Para poder usar sus bases de datos, así como la plataforma que han desarrollado, es necesario pagar una licencia que solo se concede a organizaciones (clubes, federaciones, compañías de scouting...) y a profesionales (árbitros, entrenadores, jugadores y sus agentes...). Aunque en ocasiones, ofrecen de manera gratuita, a todo el público interesado, pequeños conjuntos de datos por diversos motivos.

Durante el año 2018 *WyScout* participó en la iniciativa *Soccer Data Challenge*, ofreciendo de manera pública la mayor recopilación de datos de partidos de fútbol existente hasta la fecha. Usando este data set Luca Pappalardo, Paolo Cintia, Emanuele Massuco, Paolo Ferragina, Dino Pedreschi y Fosca Giannotti sintetizaron la información en un nuevo conjunto de datos [3], que es el empleado en este trabajo.

3.2. Contenido del conjunto de datos

El dataset que hicieron público estos investigadores contiene información sobre 7 competiciones masculinas, que son:

- **Bundesliga**: primera división del fútbol alemán.
- **LaLiga**: primera división del fútbol español.
- **Premier League**: primera división del fútbol inglés.
- **Serie A**: primera división del fútbol italiano.
- **Ligue 1**: primera división del fútbol francés.
- **Campeonato Europeo de la UEFA**: máxima competición europea de selecciones nacionales.
- **Copa Mundial de la FIFA**: máxima competición de selecciones nacionales.

La información de cada una de estas competiciones fue recopilada durante la temporada 2017-2018, a excepción del Campeonato Europeo de la UEFA que se disputó tras la temporada 2015-2016.

De cada una de estas competiciones se conoce el área geográfica en el que se disputa, el tipo de competición que es y el nombre oficial de la misma.

Además, se tienen datos de todos los partidos disputados en cada una de las ligas, diferenciándose dos temáticas principales en la información contenida en los datos: los eventos y la información de carácter más técnico.

- Eventos:
  Representan las acciones que tienen lugar durante el desarrollo de un partido, comprendiendo desde los pases hasta las diferentes decisiones que toma el árbitro. Los tipos de eventos que se contemplan en este conjunto de datos son: pases, duelos, tiros, intento de parada, interrupción, tiro libre, falta, fuera de juego, portero abandonando la línea de gol y otros en relación con el balón.

  Además, para cada uno de estos tipos de eventos, existen diferentes de sub-eventos, que los caracterizan con algo más de precisión. Para cada evento se tiene, el tipo y el instante de tiempo en el que tuvo lugar el evento, el jugador y el equipo que lo protagonizan, las coordenadas del campo en el que tiene lugar y una serie de etiquetas que aportan información extra sobre el resultado de dicho evento.

- Información técnica:
  Hace referencia al resultado final del partido, el estadio en el que se disputó, los entrenadores, las alineaciones y los cambios de los equipos que lo disputaron, los árbitros encargados de dirigirlo y la jornada o ronda de la competición en la que se encuadra.

Se cuenta también con información acerca de los jugadores, entrenadores y árbitros. De los jugadores se tiene información sobre su nombre y apellidos, lugar y fecha de nacimiento, equipo y selección nacional, en caso de ser internacional, para el que juega, altura y peso, posiciones en las que puede jugar y pierna hábil. Para los entrenadores contamos solo con su nombre y apellidos, fecha de nacimiento y nacionalidad, y equipo al que entrenan. En cuanto a los árbitros se conoce la misma información que para los entrenadores a excepción del campo indicativo del equipo al que entrenan, como es lógico.

En la misma línea, se tiene información sobre cada uno de los equipos que disputan alguna de las competiciones. La información que se tiene de cada equipo es: ciudad en la que se encuentra e información adicional sobre su área geográfica, nombre oficial, nombre oficioso y tipo (club/selección nacional).

3.3. Organización del conjunto de datos

Una vez conocemos el contenido del conjunto de datos es momento de conocer como está estructurada la información. En esta sección únicamente se va a explicar la organización de la información de interés para el desarrollo del presente trabajo. Todos los archivos en los que se organizan los datos están en formato *json*, y tienen como fuente de origen la empresa *WyScout*.

Como se ha mencionado con anterioridad, para los partidos existen dos tipos de información diferentes, los eventos y la información de carácter más técnico, y es por ello que la información de los partidos se desgrana en dos tablas diferentes, una por cada tipo de información. De esta manera, para cada competición contamos con dos archivos de datos distintos.

Los archivos que contienen la información de los eventos se denominan *events_X.json* donde X hace referencia al país en el que se disputa la competición en el caso de las ligas domésticas, y el nombre de las competiciones internacionales, *European_Championship* o *World_cup*, en caso contrario. Independientemente del tipo de competición de la que se trate, la estructura del contenido recogido es la misma. Se cuenta con tantas filas como observaciones y las siguientes columnas:

- id: identificador único de cada uno de los eventos acontecidos.
- matchId: código numérico que identifica el partido en el que ha ocurrido el evento.
- eventId: código numérico que identifica el tipo de evento del que se trata (ver Anexo A.1).
- eventName: nombre del evento que ha tenido lugar (ver Anexo A.1).
- subEventId: código numérico que identifica el tipo de sub-evento del que se trata (ver Anexo A.1).
- subEventName: nombre del sub-evento que ha tenido lugar (ver Anexo A.1).
- playerId: código numérico que identifica al jugador que ha protagonizado el evento.
- teamId: código numérico que identifica el equipo al que pertenece el jugador que ha protagonizado el evento.
- positions: coordenadas de inicio y final del evento.
- tags: códigos numéricos que identifican ciertas características que describen con mayor precisión el evento (ver Anexo A.2).
- eventSec: valor numérico que representa el instante de tiempo, medido en segundos, en el que tuvo lugar el evento.
- matchPeriod: código alfanumérico que identifica si el evento tuvo lugar en la primera o segunda parte del tiempo reglamentario, en la primera o segunda parte de la prórroga o si fue en la tanda de penaltis.

Los archivos que contienen información de carácter más técnico sobre los partidos se denominan siguiendo la misma nomenclatura que los archivos de los eventos, pero cambiando el prefijo *events* por *matches*. En este caso se tienen tantas observaciones como partidos programados en la competición, y para cada uno de ellos se tiene la siguiente información:

- status: indica si el partido ha sido jugado, *Played*, cancelado, *Cancelled*, pospuesto, *Postponed*, o suspendido, *Suspended*.
- roundId: código numérico que representa la ronda a la que pertenece el partido en la competición. En el caso de las ligas domésticas todos los partidos están bajo el mismo código, pero en las competiciones internacionales existe un código diferente para los partidos de la fase de grupos, otro para los octavos de final, otro para los cuartos de final, otro para las semifinales y uno último para la final.
- gameweek: hace referencia a la jornada de competición a la que pertenece el partido. En las ligas nacionales es un código numérico que va desde el 1, primera jornada, hasta el 38, última jornada (a excepción de la Bundesliga que tiene 36 y no 38 jornadas). Para las competiciones internacionales, este código numérico funciona exactamente igual que en las ligas domésticas en los partidos de la fase de grupos, mientras que para las fases finales este código numérico es 0.
- seasonId: valor numérico que representa la temporada a la que pertenece el partido.
- date: fecha en la que se disputó el partido en formato explícito.
- dateutc: fecha en la que se disputó el partido en formato compacto.
- winner: identificador del equipo que ganó el partido. En caso de empate este campo es '0'.
- venue: nombre del estadio en el que se jugó el partido.
- wyId: identificador único asignado al partido.
- label: contiene el nombre de los equipos que jugaron el partido y el resultado.
- referees: información sobre los árbitros que dirigieron el encuentro.
- duration: código que representa la duración del partido. En caso de disputarse en 90 minutos, entonces este campo es *'Regular'*, si se jugó prórroga entonces se tendrá el valor *'ExtraTime'*, y si acabó en penaltis entonces el valor es *'Penalties'*.
- competitionID: identificador numérico que representa la competición a la que pertenece el partido.
- teamsData: contiene múltiple información sobre cada uno de los equipos que están jugando el partido. Los campos contenidos bajo esta variable son:

- hasFormation: variable dicotómica que toma el valor 0 si no se tiene las alineaciones y los banquillos de los equipos, y el valor 1 en caso contrario.
- score: número de goles marcados por el equipo durante el partido, sin contar penaltis.
- scoreET: número de goles marcados por el equipo durante el partido, incluyendo la prórroga y sin contar penaltis.
- scoreHT: número de goles marcados por el equipo al llegar al descanso.
- scoreP: número de goles marcados por el equipo después de los penaltis.
- side: contiene el valor *'home'* si el equipo juega como local, y el valor *'away'* si juega de visitante.
- teamId: identificador del equipo.
- coachId: identificador del entrenador del equipo.
- bench: listado de jugadores que comenzaron el partido en el banquillo e información sobre su actuación en el partido: goles, goles en propia puerta y tarjetas.
- lineup: litsado de jugadores que comenzaron el partido como titulares e información sobre su actuación en el partido: goles, goles en propia puerta y tarjetas.
- substitutions: listado de los cambios realizados durante el partido del equipo en cuestión, mostrado los jugadores involucrados en esos cambios y el minuto en el que tuvieron lugar.

Además, para las competiciones de selecciones, *Copa Mundial de la FIFA* y *Campeonato Europeo de la UEFA* se cuenta con una variable más que indica, en caso de que sea un partido de fase previa, el nombre del grupo al que pertenece el partido.

Respecto a los jugadores, se tiene un único archivo con todos los jugadores que disputan alguna de las competiciones vistas, en vez de tener un archivo para los jugadores de cada competición. Este archivo se denomina *players.json* y cuenta con tantas observaciones como jugadores en el total de las 7 competiciones de interés, observándose para cada uno de ellos:

- birthArea: información geográfica sobre el lugar de nacimiento del jugador. Contiene los siguientes campos:
  - alpha2code: código de 2 caracteres que identifican el país de nacimiento.
  - alpha3code: código de 3 caracteres que identifican el país de nacimiento.
  - id: código numérico que representa el país de nacimiento.
  - name: nombre del país de nacimiento.
- birthDate: fecha de nacimiento.
- currentNationalTeamId: contiene el identificador del equipo nacional del jugador, en el caso de que juegue para el combinado nacional.

- currentTeamId: contiene el identificador del equipo al que pertenece el jugador.
- firstName: nombre de pila del jugador.
- lastName: apellidos del jugador.
- foot: pierna hábil del jugador.
- height: altura del jugador, en centímetros.
- middleName: segundo nombre, en caso de que tenga.
- passportArea: área geográfica asociada al pasaporte del jugador. Tiene la misma estructura interna que *birthArea*, aunque sus valores no tienen porque coincidir.
- role: posiciones principales dentro del esquema técnico del equipo dónde el jugador se siente cómodo. Contiene los siguientes campos:
  - code2: código alfabético de 2 caracteres que indican la posición en cuestión.
  - code3: código alfabético de 3 caracteres que indican la posición en cuestión.
  - name: nombre de la posición a la que hacen referencia los dos códigos previos.

  Existen tantas entradas para estos subcampos como posisiciones en las que el jugador pueda desempeñarse.
- shortName2: abreviatura del nombre del jugador.
- weight: peso del jugador, en kilogramos.
- wyId: identificador único del jugador.

La información que se tiene de los entrenadores se agrupa de la misma manera que la de los jugadores, es decir, en un único archivo denominado *coaches.json*. La información que se tiene de cada uno de ellos es:

- wyId: identificador único del entrenador.
- shortName: nombre por el que se conoce comúnmente al entrenador.
- firstName: nombre del entrenador.
- lastName: apellidos del entrenador.
- birthDate: fecha de nacimiento.
- birthArea: información sobre el área geográfica del lugar de nacimiento del entrenador. Contiene los mismos subcampos que esta misma variable recogida para los jugadores.

- passportArea: área geográfica asociada al pasaporte del entrenador. De nuevo, contienen los mismos subcampos que esta misma variable recogida para los jugadores.
- currentTeamId: identificador del equipo al que dirige.

La información de los equipos involucrados en alguna de las siete competiciones mencionadas está contenida en el fichero *teams.json*, y para cada equipo contiene la siguiente información:

- city: ciudad en la que se ubica el equipo.
- name: nombre con el que se conoce comúnmente al equipo.
- area: área geográfica en la que se ubica el equipo. Contiene los mismo campos que los vistos para la variable *birthArea* en la información de los jugadores y entrenadores.
- wyId: identificador único del equipo.
- officialName: nombre oficial del equipo.
- type: campo que en el caso de tratarse de un club contiene el valor *'club* y en caso de ser un combinado nacional contiene *'national'*

Teniendo en cuenta lo anterior, este TFG se realizará a partir de los ficheros:

- events_X.json.
- matches_X.json.
- players.json
- teams.json

3.4. Preprocesamiento de los datos

Para poder utilizar la información contenida en estos ficheros, de una forma más sencilla y cómoda, se ha sometido a cada uno de ellos a distintos tratamientos.

El fichero *teams.json* no sufre ninguna modificación en su estructura interna, pero se cambia su formato a *.csv*, que será el que usaremos para todos los ficheros nuevos.

El fichero de jugadores, *players.json*, tras ser leídos convenientemente con el paquete *jsonlite* [14] de R, y haciendo uso también de la librería *dplyr* [15], fue convertido a un fichero de tipo *.csv*. Además, se generó una columna adicional denominada *league*, que en caso de que el jugador pertenezca a un club del sistema de competición de algunas de las grandes cinco ligas europeas, indica en cual de ellas lo hace. En caso contrario, contiene un valor *NA*.

Los archivos *events_X.json* que contienen los eventos de los partidos agrupados por competición, tienen 12 columnas distintas de las cuales dos de ellas, *positions* y *tags*, contienen un conjunto de valores en vez de un único valor. Con el fin de poder acceder a estos valores de una manera más cómoda se realizan los siguientes procesos para cada columna:

- positions: esta columna contiene dos pares de coordenadas que hacen referencia a la posición en el campo en el que comienza el evento, (x1,y1), y la posición en la que finaliza, (x2, y2). Tras el proceso, se crean 4 nuevas columnas, (x1, y1, x2, y2), de tal forma que cada una de ellas contiene uno de los cuatro valores diferentes que antes se almacenaban en una lista.

- tags: esta columna almacena para cada evento un listado de hasta seis identificadores numéricos que aportan información extra sobre el evento en cuestión (ver Anexo A.2). Al igual que ocurre con la columna *positions*, cada uno de estos valores pasa a ser parte de una nueva columna, por lo que se han generado un total de seis columnas nuevas de tal manera que cada una de ellas contiene uno de los identificadores de la lista. Las nuevas columnas se denominan: *tags_id1*, *tags_id2*, *tags_id3*, *tags_id4*, *tags_id5* y *tags_id6*.

Una vez hemos tratado la columna *positions* es el momento de adaptar sus valores. Según podemos ver en el glosario de *WyScout* [16], los valores de x e y están medidos en el intervalo [0,100]. La posición es medida siempre desde la perspectiva del equipo poseedor del balón o protagonista del evento, siendo la x la longitud y la y la anchura. De esta manera, los valores bajos de la x reflejan la cercanía al área propia (valores bajos) o a la rival (valores altos), mientras que la y representa el flanco del campo en el que acontece el evento, siendo la banda izquierda la que se corresponde con valores cercanos al 0 y la derecha los próximos al 100.

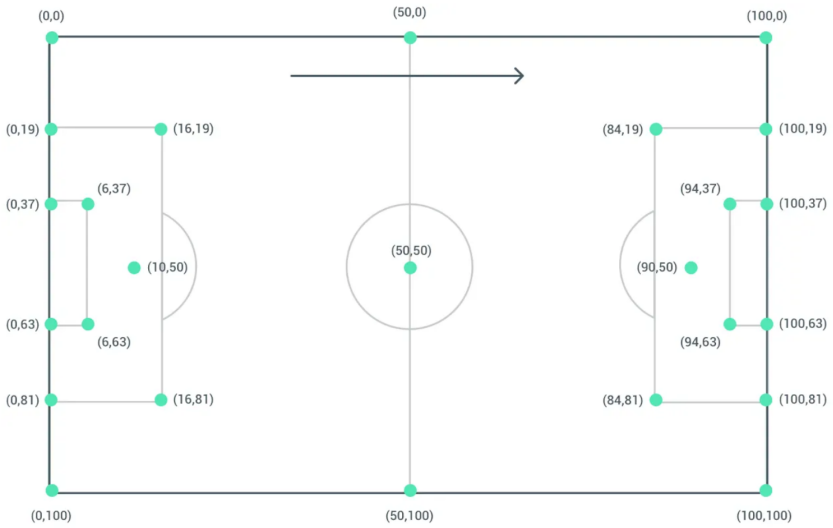

Figura 5: Sistema de coordenadas de *WyScout*.

Estos valores son necesarios adaptarlos a las dimensiones reales de un terreno de juego, ya que si se consideran las coordenadas proporcionadas tendríamos un cuadrado de 100x100. *FIFA* establece que la longitud del campo, en metros, ha de estar comprendida en el intervalo [90,120], mientras que la anchura puede variar en [45,90]. Para el desarrollo de este trabajo se ha tomado como referencia el estándar 105x68 metros, adaptando convenientemente las coordenadas a estas nuevas medidas.

Además, se ha generado una nueva columna denominada *competition* que identifica cada evento a la competición a la que se corresponde. Así, por ejemplo, un evento que ocurre en un partido de la liga alemana tendrá en este nuevo campo el valor *'Germany'*. Por último, se han juntado los eventos de las seis competiciones en un mismo conjunto de datos que se ha denominado *events* y que se ha guardado bajo el formato *csv*.

Para los ficheros *matches_X.json* se han mantenido todas las columnas originales excepto *teamsData* y todas las que penden de ella. Además, para los ficheros correspondientes a las ligas, se ha generado la columna *groupName* que originalmente sólo figura en *matches_World_Cup.json*. Tras esto, los seis ficheros se agrupan en el fichero *matches* en formato *csv*, como todos los demás.

Tras todos estas modificaciones obtenemos un total de 4 ficheros, *teams.csv*, *players.csv*, *matches.csv* y *events.csv*, cuyas estructuras son prácticamente idénticas a las originales.

4. Contextualización

En este capítulo, primeramente, se introduce el concepto de goles esperados, o *expected goals*, así como su utilidad y relevancia para la construcción de otras métricas de interés en el mundo del fútbol. En segundo lugar, se exponen los motivos por los que se ha considerado que las primeras divisiones de *Alemania*, *España*, *Francia*, *Inglaterra* e *Italia*, de la temporada *2017-2018*, constituyen una muestra representativa para la construcción de un modelo de goles esperados para el Mundial de Rusia del 2018.

4.1. La Estadística en el mundo del deporte

El mundo del deporte y el de los datos, y la información, siempre han estado muy ligados, aunque es en los últimos veinte años dónde la relación entre ambos se ha ido estrechando cada vez más.

En el año 2001, Billy Beane, entrenador de los *Oakland Athletics* de la liga americana de béisbol, junto a Paul dePodesta, un joven economista, cambiaron para siempre el uso de la Estadística en el ámbito deportivo. Desarrollaron un nuevo sistema de valoración de jugadores que permitía detectar a aquellos que estaban siendo infravalorados por las métricas existentes hasta el momento.

Gracias a este nuevo modelo, pudieron reconstruir su plantilla de forma exitosa, convirtiéndose en uno de los equipos punteros de la competición, siendo una de las franquicias con menos recursos disponibles, e incluso batir récords de imbatibilidad a lo largo de la temporada. Finalmente, no consiguieron hacerse con el campeonato, pero su método de trabajo basado en la Estadística sentó un antes y un después en el mundo del deporte.

Inspirados por este hecho, los clubes de fútbol de alto nivel, como otras muchas entidades deportivas, han ido incorporando poco a poco profesionales de la Estadística en sus organigramas, con el objetivo de lograr los mejores resultados posibles con los recursos de los que disponen.

Pero no sólo son los clubes los que utilizan la Estadística avanzada para medir el rendimiento de jugadores y equipos, sino que cada vez es más frecuente encontrarse con métricas de este estilo en los análisis deportivos realizados desde los medios de comunicación. Como consecuencia, cada vez es mayor el número de aficionados que se sienten familiarizados con estas nuevas métricas, siendo un claro ejemplo de ello los *goles esperados (expected goals)*.

4.1.1. ¿Qué son los goles esperados?

Los goles esperados, o xG (abreviatura de *expected goals*), miden la cantidad de goles que un equipo o jugador deberían haber logrado en función de la calidad de las ocasiones de las que dispuso a lo

largo de un partido, competición o temporada. Para ello, cada ocasión de gol es convenientemente evaluada y se le asigna la probabilidad de finalizar en gol en función de las circunstancias en las que se dió. Por ejemplo, una ocasión con una probabilidad de 0.5 significa que si esa misma situación se diese 100 veces acabaría en gol en 50.

A pesar del gran auge que ha adquirido esta métrica en los últimos cinco años realmente sus inicios se remontan a casi 30 años atrás. En 1993 *Vic Barnett* y *Sarah Hilditch* fueron los primeros investigadores que hicieron referencia al término *'goles esperados'* en un estudio sobre el impacto del rendimiento de los equipos que disputaban sus partidos como local sobre hierba artificial [17].

No fue hasta el año 2004 cuando se comenzó a hablar de los *goles esperados* bajo la misma concepción que en la actualidad. Por un lado, los investigadores *Jake Ensum*, *Richard Pollard* y *Samuel Taylor* publicaron un estudio [18] basado en 37 partidos disputados durante el *Mundial de Corea (2002)* en el que desarrollaron modelo de regresión logística que logró identificar hasta 5 variables distintas que afectaban a la probabilidad de que un tiro finalizase en gol:

- Distancia a la portería desde la posición de golpeo.
- Ángulo respecto a la portería desde la posición de golpeo.
- Existencia de un defensor a menos de un metro desde la posición de golpeo.
- Si la acción previa al golpeo era un centro o no.
- Número de jugadores entre el balón y la portería en el momento del golpeo.

Por otro lado, *Alan Ryder* compartió con la comunidad científica una metodología [19] para el análisis de la calidad de los tiros en *hockey hielo*, basada en los siguientes pasos:

- Recolectar los datos y analizar la probabilidad de acabar en gol para cada circunstancia de tiro.
- Construir un modelo para calcular la probabilidad de acabar en gol en función de las circunstancias propias del tiro en cuestión.
- Determinar, para cada tiro, la probabilidad de terminar en gol.
- Definir los goles esperados como la suma de las probabilidades de todos los tiros de dicho jugador, partido o competición.
- Calcular los goles esperados normalizados.

En el año 2009 Howard Hamilton [20] fue un paso más allá y propuso un modelo capaz de evaluar las diferentes jugadas de un partido y proporcionar para cada una de ellas la probabilidad de acabar en gol. Un enfoque algo distinto al concepto de *'goles esperados'* que se emplea en este trabajo pero

que sirvió como impulso para otras métricas como los *player ratings* basados en *VAEP* (*Valoración de las Acciones según la Estimación de Probabilidades*).

Sam Green, en el año 2012, realizó la última gran aportación hasta el día de hoy a los modelos de *'goles esperados'*. Su estudio [21] basado en tratar de descubrir las zonas del campo en las que es más verosímil que un tiro acabe en gol obtuvo grandes resultados y este hecho impulso el uso de esta métrica que a día de hoy es una de las más famosas e interesantes para los aficionados.

4.1.2. Usos de la métrica goles esperados

Al ser una métrica basada en un concepto sencillo y fácil de entender por el público en general y a su vez estar relacionada con uno de los aspecto más importantes en el mundo del fútbol, como son los goles, se le puede encontrar una gran variedad de usos distintos, incluso a diferentes niveles de análisis.

El uso más popular es el que se le da en los **análisis de prensa deportiva** en el que se establecen comparaciones entre los equipos que van a disputar un partido y se hace referencia a los goles esperados de cada uno de estos equipos a lo largo de la competición en la que se disputa el partido. Establece un buen punto de comparación entre los contrincantes ya que permite hablar de la capacidad ofensiva de cada uno de ellos sin tener que profundizar en tácticas y/o análisis algo más exhaustivos en los que el lector puede sentirse incómodo. Si sirve para realizar la comparación entre ambos equipos antes del enfrentamiento, también sirve para hacer lo propio una vez disputado el partido por lo que su uso también es muy frecuente en las crónicas post-partido.

Alejándonos del público en general y adentrándonos un poco más en el profesionalismo nos encontramos con **usos algo más avanzados y/o con valor añadido para técnicos, scouts y directores deportivos**. Dentro de este ámbito podemos encontrarnos con dos perspectivas distintas pero con necesidades mutuas entre ellas, que son: análisis individual de un jugador y análisis del juego de un equipo.

Si nos centramos en el análisis individual de un jugador podemos usar la métrica de los goles esperados para conocer su fiabilidad de cara a puerta. Por ejemplo, el delantero centro del Real Valladolid Sergi Guardiola acumuló durante la pasada temporada, 2020-2021, un total de 5.0 goles esperados en los 29 partidos que disputó, anotando únicamente un gol [22]. Sin embargo, su compañero Shon Weissman en 32 partidos acumuló un total de 5.3 goles esperados, consiguiendo hasta 6 goles [22]. De esta manera, podemos ver la capacidad de definición de cada uno de estos dos jugadores y asegurar que la campaña de Weissman ha sido mucho mejor en el aspecto definitorio que la de Guardiola.

Además de establecer comparaciones entre diferentes jugadores también se puede emplear para comparar la evolución de un mismo jugador a lo largo de diversas temporadas y ver si sus números de cara a puerta han ido mejorando o empeorando con el paso del tiempo. Por ejemplo Kike García, el actual delantero de Osasuna, durante la campaña 2020-2021 vivió un idilio con el gol consiguiendo

12 tantos habiendo acumulado un total de 10.7 goles esperados. Se trata de la única campaña de las últimas cuatro, jugadas todas ellas en el Eibar, en la que consiguó más goles que los que cabrían esperar y por ello podemos pensar que realmente fue una excepción y que lo más lógico es que el año que viene su capacidad anotadora no sea tan elevada cómo la de este año.

En definitiva, la métrica de goles esperados permite establecer una **comparación** de la capacidad goleadora de los **jugadores** de una manera más realista que el número total de goles marcados. Por ejemplo, si comparamos los registros goleadores de Antoine Griezmann con los de Borja Iglesias en la temporada 2020-2021 podemos ver que Antoine consiguío 2 goles más que Borja Iglesias, pero su rendimiento cara a puerta fue inferior al de Borja. El primero presenta un saldo de 'goles marcados'-'goles esperados' de -0.9, mientras que para el segundo asciende a +3.3 [22].

Desde el punto de vista del análisis del juego de un equipo podemos utilizar la métrica de goles esperados para hacernos una idea del **estilo de juego de un equipo**. Es esperable que los equipos que desarrollan un juego con una vocación más ofensiva que defensiva tengan a lo largo de toda una temporada un número de goles esperados mayor que un equipo que base su juego en la solidez defensiva. Así, por ejemplo, podemos establecer una comparación entre el Atletico de Madrid y el FC Barcelona [22].

Por un lado, el Atlético de Madrid, desde la llegada de su entrenador, Diego Simeone, en 2012, se ha caracterizado por ser un equipo que basa sus éxitos en una buena defensa lo que le permite ganar partidos por estrechas diferencias en el resultado. Por otro lado, el FC Barcelona siempre se ha caracterizado por un juego basado en posesiones largas que le permiten atacar la meta rival y a su vez defenderse de posibles ataques rivales. Si nos fijamos en los goles esperados para ambos equipos podemos ver cómo los respectivos valores reflejan un **mayor caudal ofensivo** por parte del FC Barcelona, con 78.9xG, que por parte del Atlético de Madrid, 52.4xG.

Desgranando el valor de los goles esperados en diferentes intervalos de tiempo durante los partidos podemos ver **patrones de conducta e intensidad en la producción ofensiva**. Estas diferencias por intervalos pueden ser más evidentes en los últimos partidos de una competición, en los que se tiende a ser más o menos conservador en función de los objetivos propuestos, pero aún así se pueden detectar diferencias significativas a lo largo de toda la temporada. También se puede utilizar esta comparación por intervalos de tiempo condicionando al hecho de que el equipo actúe como local o lo haga como visitante. Existen muchos equipos que varían su plan de juego en función de su condición de local o visitante por lo que es lógico esperar que esto se refleje en la distribución temporal de los goles esperados.

4.1.3. Limitaciones de la métrica goles esperados

A pesar de ser de una métrica de gran utilidad y de gozar de una creciente popularidad también cuenta con una serie de factores que condicionan las interpretaciones que se pueden extraer de ella. El factor más importante a tener en cuenta a la hora de hablar de la métrica de goles esperados es que a diferencia de otras estadísticas habituales como la posesión, el porcentaje de pases precisos o

el número de faltas realizadas, los valores de goles esperados, así como las probabilidades asignadas a cada tiro, son obtenidos a partir de un modelo estadístico. Por ello, no debe darse un uso muy protagonista de esta métrica para analizar un único partido, o un pequeño conjunto de ellos, ya que para pequeñas muestras se comporta de manera algo imprecisa. Sin embargo, si se emplea con un conjunto muestral grande, la precisión de los resultados totales será muy elevada.

Hay que tener en cuenta que, al no ser datos directamente cuantificables como las métricas tradicionales, podemos encontrarnos con distintas evaluaciones para un mismo tiro, partido, equipo o competición. En la mayoría de las ocasiones los proveedores de este tipo de métricas cuentan con su propio modelo y, por tanto, la evaluación de un mismo tiro u ocasión puede diferir ampliamente entre las distintas compañías.

Las diferencias entre los modelos de los distintos proveedores se pueden esquematizar en dos grupos temáticos: datos empleados y características propias del modelo.

A la hora de hablar de los datos empleados es importante distinguir entre los proveedores de las métricas y los proveedores de los datos. Los primeros son aquellos servicios en los que podemos encontrar esta y muchas otras métricas sobre jugadores, equipos y competiciones, mientras que los segundos son todas aquellos grupos empresariales encargados de recopilar toda la información que se produce durante un partido.

La extracción de la información generada durante el encuentro y su posterior conversión en datos es un proceso en el que participan tanto programas informáticos como personas, lo que aporta cierto sesgo de subjetividad que puede provocar grandes diferencias en los resultados obtenidos. Por ejemplo, una compañía puede considerar que cierto tiro o jugada no supuso una gran ocasión mientras que en una segunda compañía podemos encontrarnos que dicha jugada fue una ocasión clara y manifiesta de gol. Por ello, es muy importante a la hora de hablar de goles esperados tener en cuenta el proveedor de los datos que se emplea en dicho modelo y tratar de evitar establecer comparaciones entre métricas basadas en datos de distintas compañías.

Que dos compañías compartan mismo proveedor de información no quiere decir que hagan un mismo uso de la misma, por lo que también es frecuente encontrar diferencias entre los modelos generados con datos de una misma fuente. Aunque dispongan de los mismos datos, las características propias del modelo desarrollado no tiene porque ser las mismas y, por tanto, los resultados pueden diferir de unos modelos a otros.

Los aspectos más relevantes a considerar en los modelos son:

- **Marco teórico del modelo**: En la actualidad existen múltiples opciones para implementar un modelo de estas características, partiendo de las más básicas como una *regresión logística* hasta el uso de árboles de decisión en técnicas de *machine learning* como el *gradient boosting*.
- **Variables consideradas**: No todos los modelos tienen porque incluir las mismas variables, ni tampoco estas tienen porque ser medidas y procesadas de igual forma.

Como hemos visto hasta ahora, la métrica de goles esperados sirve para cuantificar la probabilidad de que cierto tiro o jugada acabe en gol en función de acciones previas similares. Para definir la similaridad entre las ocasiones se puede emplear la posición en el campo y el ángulo respecto a la portería desde dónde se realiza el tiro, el número de defensas a menos de un metro del balón, etcétera, pero en ningún momento se tiene en cuenta qué jugador lo protagoniza. Por ello, todos los cálculos están realizados para un jugador promedio. Esto supone un problema para aquellos jugadores para los que su capacidad goleadora está por encima, o por debajo, de la media. Por ejemplo, la probabilidad de que un tiro desde la frontal del área acabe en gol no puede ser la misma si la jugada la protagoniza Leo Messi o Karim Benzema, que tienen una capacidad anotadora elevada, que si el protagonista es Vinícius Júnior, cuyo acierto de cara a puerta es muy bajo. Tanto en un caso como en otro la métrica de goles esperados no hace justicia a la realidad ya que asignará en ambos escenarios la misma probabilidad de acabar en gol.

4.1.4. Otras métricas basadas en los goles esperados

La métrica de goles esperados puede usarse para generar otras métricas que ayuden a una mejor interpretación de su valor, como puede ser los *goles esperados por tiro efectuado* o los *goles esperados sin considerar penaltis*.

Estas dos métricas pueden ser de utilidad para comparar el rendimiento de dos jugadores. Supongamos la situación ficticia para los jugadores A y B: El jugador A ha convertido 15 goles teniendo un total de goles esperados igual a 15, mientras que el jugador B con la misma cantidad de goles esperados, solo ha logrado 10 tantos. Utilizar las dos métricas mencionadas, junto a sus goles esperados, seremos capaz de hacer una valoración más justa.

Existen otras métricas que pueden generarse a partir de los goles esperados, y que no tienen por qué medir el rendimiento del jugador que efectúa el tiro. Algunas de ellas son: *asistencias esperadas*, *goles esperados por posesión* y *goles posteriores al tiro*.

A continuación, se va a realizar una breve exposición sobre las cinco métricas mencionadas.

- **Goles esperados por tiro efectuado (xG/S)**:
  Se calcula dividiendo el total de goles esperados para un jugador entre el número de tiros realizados. Retomando nuestro ejemplo ficticio, imaginemos que el jugador B tiene un xG/S de 0.13 mientras que para A, es igual a 0.22, como puede observarse en el Cuadro 1.

Jugador	**Goles**	**xG**	**xG/S**
A	15	15	0.22
B	10	15	0.13

Cuadro 1: Comparación de jugadores A y B mediante sus xG y xG/S.

Al acumular ambos jugadores un mismo número de goles esperados, y tener el jugador B un xG/S menor que el de A, esto quiere decir que para llegar hasta esos 15 goles esperados, el jugador B tuvo que protagonizar muchas más jugadas que el jugador A y, por tanto, la probabilidad de finalizar en gol de dichas oportunidades era bastante más baja que en el caso de A. Bajo este escenario, no parece fiable afirmar que la capacidad de definición de B es peor que la de A. También podría haberse dado el escenario opuesto, y que el xG/S de A fuese inferior al de B lo que si que significaría que A es mejor definidor que B.

- **Goles esperados sin penaltis (npxG)**:
  Retomando la comparación de nuestros jugadores ficticios, A y B, podemos ponernos en la situación de que el jugador A logró 5 de sus goles desde el punto de penalti, mientras que el jugador B no lanzó ninguno. En este nuevo escenario lo más justo sería poder establecer una comparación entre ambos jugadores pero bajo los mismos condicionantes, es decir, eliminando del cómputo las acciones desde los 11 metros. Para ello, surge la métrica de *goles esperados sin contabilizar penaltis*, npxG.

Jugador	**Goles**	**Goles sin penaltis**	**npxG**
A	15	10	10.44
B	10	10	15

Cuadro 2: Comparación de jugadores A y B mediante sus xG y xG/S.

Suponiendo una probabilidad fija para los penaltis de acabar en gol de 0.76, los datos para nuestros jugadores son los que se muestran en el Cuadro 2. Bajo esta nueva métrica tenemos que A consiguió 10 goles y se esperaban de él 10.44 mientras que B logró también 10 pero se esperaban de él 15. Al igual que hicimos con los xG, podríamos calcular el cociente npxG/S para una mejor comparación de rendimiento.

- **Goles esperados por posesión (xGPos)**:
  Esta nueva métrica calcula la probabilidad de que una jugada entera acabe con éxito. Para determinar el xGPos de una jugada se ha de usar los xG de todos los tiros que tuvieron lugar en esa jugada y emplearlos para calcular la probabilidad de que el equipo rival no conceda el gol.

  Por ejemplo, si en una jugada el equipo atacante realiza 3 disparos con xG asociados de 0.25, 0.54 y 0.78, la probabilidad de que el equipo rival no conceda un gol en esa jugada es de: (1-0.25) x (1-0.54) x (1-0.78) = 0.0759. Por tanto, la probabilidad de que esa jugada acabe en gol para el equipo atacante es 1 - 0.0759 = 0.9241.

- **Asistencias esperadas (xA)**:
  Si en vez de medir la capacidad anotadora de un jugador queremos medir su capacidad para

generar ocasiones de gol, podemos recurrir a la métrica de *asistencias esperadas*, o xA. Se sirve de los goles esperados para obtener el xA de un determinado jugador. Para ello, a cada jugador se le asigna un xA igual a la suma de los xG de los tiros precedidos por una asistencia suya.

De esta manera, si un jugador tiene un xA de 7, quiere decir que a través de sus pases ha generado ocasiones suficientes como para conseguir 7 goles. Esta métrica nos puede ayudar a saber si las cifras de un jugador se están viendo perjudicadas, o favorecidas, por el rendimiento de sus compañeros. Si un jugador tiene un xA de 7 y realmente sólo ha dado 3 asistencias, eso quiere decir que sus compañeros están rindiendo por debajo de la media. También podría darse el caso contrario en el que el jugador acumula un xA de 7 y realmente ha dado 10 asistencias, por lo que sus compañeros están ayudando a que sus cifras sean mejores.

- **Goles esperados posteriores al tiro (PSxG)**:
  Esta métrica sirve para medir el rendimiento de los porteros. La diferencia que presenta frente a los goles esperados, es que únicamente se consideran con una probabilidad mayor que 0 a aquellos tiros que van a puerta. De esta forma, podemos medir el rendimiento de un arquero comparando la suma de los PSxG efectuados sobre su portería y el número real de goles encajados.

En este TFG, para generar el modelo de predicción de goles esperados, se van a utilizar los datos de las cinco grandes ligas europeas durante la temporada 2017-2018, temporada previa a la disputa del mundial celebrado en Rusia.

4.2. Las cinco grandes ligas como espacio muestral

A lo largo de esta sección se va a justificar por qué se ha considerado que las cinco grandes ligas europeas suponen una buena representación para la creación del modelo de goles esperados para el Mundial de Rusia de 2018. Recordemos que estas competiciones son:

- *Bundesliga*, 1ª División de Alemania.
- *LaLiga*, 1ª División de España.
- *Ligue 1*, 1ª División de Francia.
- *Premier League*, 1ª División de Inglaterra.
- *Serie A*, 1ª División de Italia.

4.2.1. Relevancia de estas competiciones en el Mundial de Rusia 2018

La FIFA, *Fédération Internationale de Football Association*, es el máximo organismo del fútbol mundial y es el organizador de la Copa Mundial de selecciones. Este organismo está conformado por diferentes organismos reguladores del fútbol en los diferentes continentes y áreas socio-económicas de importancia. El mapa mundial futbolístico está compuesto por 6 áreas de importancia, también denominadas confederaciones, con sus respectivas organizaciones al mando. Las confederaciones son:

- Confederación Sudamericana de Fútbol (CONMEBOL): máximo organismo en América del Sur.
- Unión de Asociaciones Europeas de Fútbol (UEFA): máximo organismo en Europa.
- Confederación Asiática de Fútbol (AFC): máximo organismo en Asia.
- Confederación Africana de Fútbol (CAF): máximo organismo en África.
- Confederación de Fútbol de Norte, Centroamérica y el Caribe (CONCACAF): máximo organismo en América del Norte, América Central y el Caribe.
- Confederación de Fútbol de Oceanía (OFC): máximo organismo en Oceanía.

La FIFA, como organizador de la Copa Mundial, establece cuántos combinados nacionales de cada confederación compiten en el torneo. Para el año 2018, los cupos por confederaciones fueron los que se muestran en el Cuadro 3.

CONFEDERACIÓN	**Nº PLAZAS**
CONMEBOL	4.5
UEFA	13
AFC	4.5
CAF	5
CONCACAF	3.5
OFC	0.5

Cuadro 3: Cupos por federación

Existen confederaciones que tienen un número no entero de plazas otorgadas, esto se debe a las denominadas *'repescas'*. En cada confederación existe una fase de clasificación disputada entre los países que la conforman y una vez finalizada y otorgadas las plazas para el mundial existen unas eliminatorias entre las mejores selecciones, de diferentes confederaciones, que no hayan conseguido una plaza en su fase de clasificación. En esta ocasión el combinado de Australia, *AFC*, se enfrentó al de Honduras, *CONCACAF*, y el de Nueva Zelanda, *OFC*, se enfrentó al de Perú, *CONMEBOL*. En

estas eliminatorias salieron vencedores Australia y Perú, por lo que finalmente la distribución de las plazas, por confederaciones, y teniendo en cuenta que el anfitrión, Rusia, tenía plaza asegurada, fue la mostrada en el Cuadro 4.

CONFEDERACIÓN	**Nº SELECCIONES**	**% PARTICIPANTES**
CONMEBOL	5	15.625
UEFA	14	43.75
AFC	5	15.625
CAF	5	15.625
CONCACAF	3	9.375
OFC	0	0.0

Cuadro 4: Selecciones participantes por confederación

En total, el **43.75 %**, de las selecciones nacionales que participaron en este mundial pertenecían a la *UEFA* mientras que las tres siguientes confederaciones con mayor peso en el torneo, *CONMEBOL*, *AFC* y *CAF*, significaban, de manera conjunta, el **46.875 %** de las selecciones participantes.

La relevancia del fútbol europeo en el *Mundial de Rusia* no sólo se dió en el número de selecciones del viejo continente que participaron, sino que las ligas domésticas de estos países fueron las que más jugadores aportaron al campeonato.

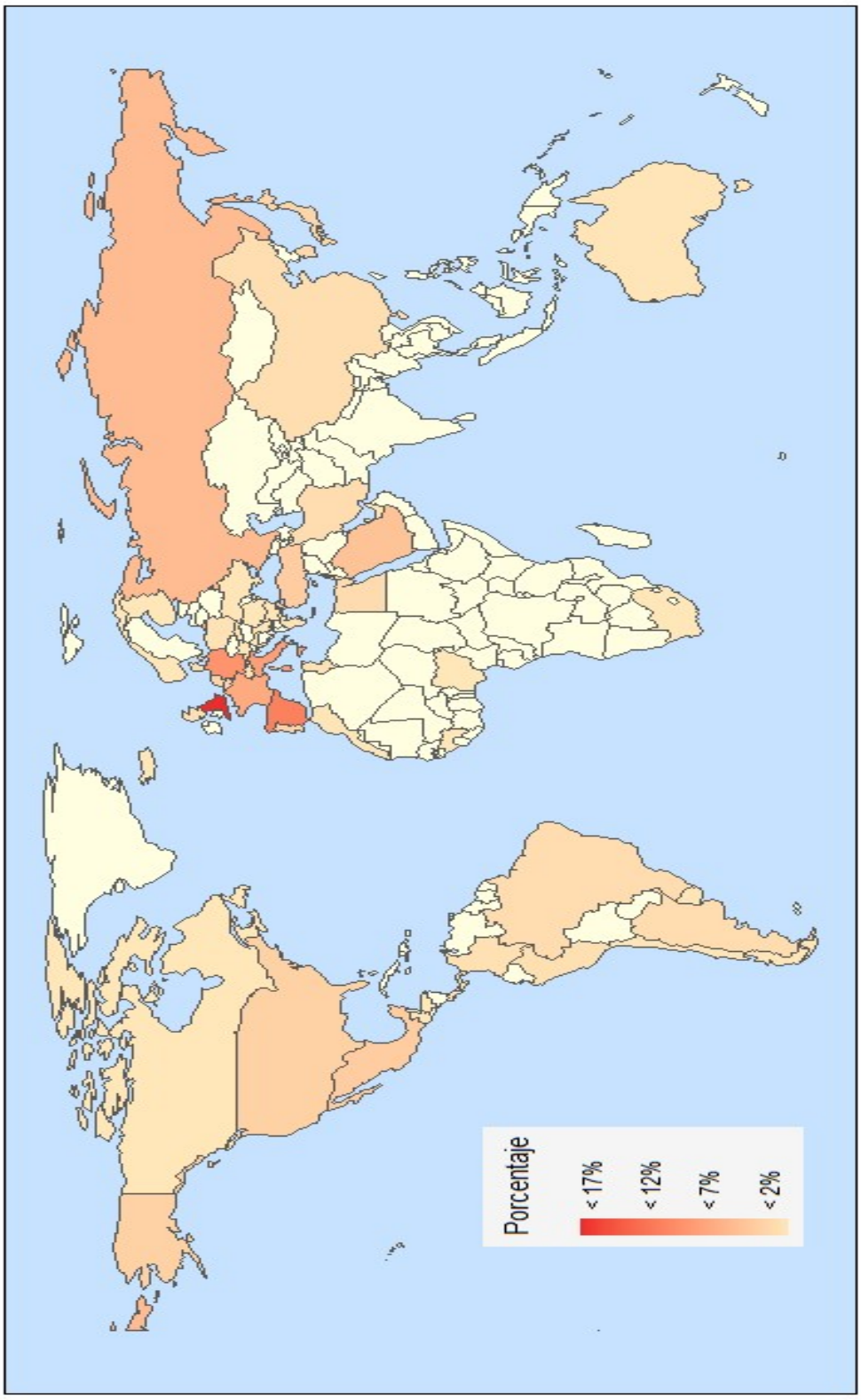

Figura 6: Porcentaje de jugadores aportados por las ligas de cada país

En la Figura 6, (ver Anexo B.1), hemos representado la influencia en el torneo de las ligas de los distintos países, en función del número de jugadores que aportaron. Los colores más oscuros se corresponden con aquellos países que concentran un mayor tanto por ciento de jugadores convocados jugando en sus ligas, mientras que los colores más claros se corresponden con países en cuyas ligas apenas hay jugadores mundialistas. El continente europeo es dónde encontramos la mayor parte de las zonas más oscuras del mapa lo que nos indica que estas ligas cuentan con muchos más jugadores participando en el mundial que las demás.

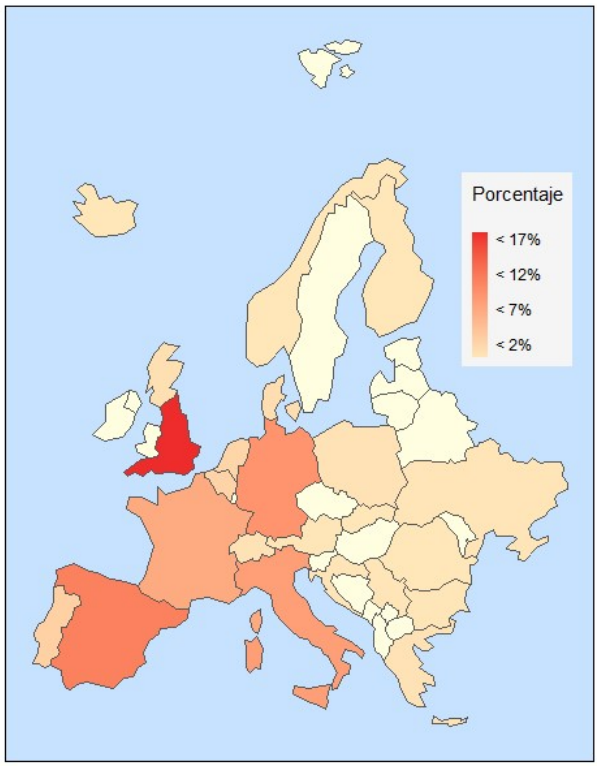

Figura 7: Porcentaje de jugadores aportados por las ligas europeas.

En la Figura 7, (ver Anexo B.1), es más sencillo observar la influencia de las ligas de cada uno de los países europeos. Inglaterra es el país que mayor aportación de jugadores hace al torneo, con un porcentaje, sobre el total, superior al 17.66 %. En segunda posición tenemos a España, con un porcentaje del 11 %. A continuación, en el siguiente nivel de aportación, contamos con las ligas de Alemania e Italia con porcentajes bastante similares, 9.10 % y 7.88 % respectivamente. En un escalón por debajo se encuentra Francia, aportando el 6.66 % de los jugadores. Sumando los porcentajes de estos cinco países nos damos cuenta de que el **52.30 %** de los jugadores de la copa mundial jugaron ese año en alguna de las ligas de Inglaterra, España, Alemania, Italia y Francia.

Es tanto el nivel de importancia de estos países en el ámbito futbolístico que incluso sus segundas y terceras divisiones aportan jugadores a las diferentes selecciones del torneo (ver Anexo B.1).

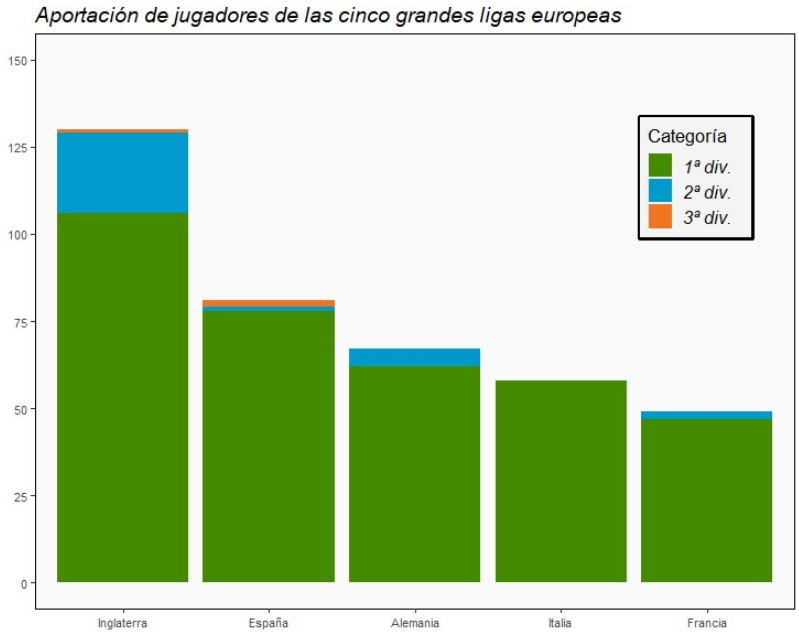

Figura 8: Distribución de los jugadores en las diversas categorías.

Cierto es que la mayor parte de los jugadores juegan en las primeras divisiones de dichos países pero aún así el porcentaje de jugadores que juegan en categorías de menor nivel no es nada desdeñable. Destaca, sobre todo, la aportación de la segunda división inglesa, con un total de 23 jugadores, que es la misma cantidad de jugadores que puede llevar una selección al torneo. Además, la aportación de la segunda categoría del fútbol inglés sólo es superada por estos cinco países, *Rusia* y *Arabia Saudi*.

Italia es el único de los cinco países de referencia que sólo aporta jugadores desde su primera división. Aún así, *Italia* sigue siendo el cuarto país que más jugadores del mundial tiene jugando en su liga, algo más meritorio aún si cabe si se tiene en cuenta que su selección nacional no disputó el torneo.

Si sólo contamos los jugadores provenientes de estas cinco primeras divisiones el porcentaje que representan sobre el total de jugadores que disputan el mundial es del **47.69 %**, prácticamente la mitad de los jugadores que fueron convocados para el torneo.

Negar la relevancia de estos cinco países en la influencia del desarrollo del juego es prácticamente imposible, por lo que trabajar con los datos de los partidos de estas competiciones nos permite partir de una muy buena aproximación para los análisis posteriores.

4.2.2. Diferencias y semejanzas entre ellas

En este capítulo se va a tratar de establecer una comparación entre las cinco grandes ligas europeas (Alemania, España, Francia, Inglaterra e Italia), en función de las nacionalidades de sus jugadores y

de la evolución de los resultados de los partidos.

Históricamente entre estas competiciones existían más diferencias en el estilo de juego que las que se pueden encontrar en la actualidad. Estas diferencias se han reducido considerablemente cómo consecuencia del fenómeno de la *globalización*, que ha permitido, a los clubs y técnicos, analizar a cualquier rival e incorporar aspectos del juego de cualquiera de ellos. Además, con la llegada de la *globalización* llegó el denominado *fútbol moderno*, en el que las ingentes cantidades de dinero que se mueve ha repercutido también en la mentalidad de jugadores y técnicos, que en su mayoría no sienten un arraigo tan grande por su club de formación, lo que les hace cambiar de destinos con una mayor frecuencia que antes.

Si nos fijamos en las nacionalidades de los jugadores que compitieron en estas ligas durante la temporada 2017-2018 podemos ver que únicamente *LaLiga* y la *Ligue 1* cuentan con más de un 50 % de jugadores nacionales (ver Cuadro 5).

Competición	**Nº jugadores**	**Nº nacionales**	**% nacionales**
LaLiga	503	295	58.65
Premier League	442	162	36.66
Serie A	520	226	43.46
Ligue 1	506	290	57.31
Bundesliga	468	232	49.57

Cuadro 5: Jugadores nacionales en cada liga doméstica

El porcentaje más bajo de jugadores nacionales lo encontramos en la *Premier League*, con un 36.66 %, seguido por la *Serie A* en el que cuatro de cada diez jugadores es nacido en Italia. La *Bundesliga* acaricia la mitad de jugadores nacionales, pero se queda en un porcentaje del 49.57 %. Si en vez de contabilizar para cada competición el número de jugadores nacionales, contamos el número de jugadores nacidos en alguno de los otros cuatro países de las grandes ligas, obtenemos los valores que se muestran en el Cuadro 6.

Competición	**Nº jug.**	**Nº nac. OGL**	**% nac. OGL**
LaLiga	503	34	6.76
Premier League	442	79	17.87
Serie A	520	51	9.81
Ligue 1	506	15	2.96
Bundesliga	468	43	9.19

Cuadro 6: Jugadores nacidos en uno de los cinco países y jugando en otro.

La *Premier League* vuelve a ser la competición que más destaca, en este caso por ser la que mayor porcentaje de jugadores provenientes de los otros cuatro países tiene en sus clubs con un 17.87 %. Las

dos siguientes competiciones que mayor porcentaje acumulan son la *Serie A* y la *Bundesliga* con algo más de la mitad que la *Premier League*, siendo los porcentajes de 9.81 % y 9.19 % respectivamente. Por último tenemos a *LaLiga* con el 6.76 % y muy por detrás la *Ligue 1* con prácticamente el 3 % del total de jugadores.

Considerando para cada liga tanto los jugadores nacionales, Cuadro 5, como los nacidos en alguno de los otros cuatro países, Cuadro 6, obtenemos los resultados mostrados en el Cuadro 7.

Competición	**Nº jug.**	**Nº nac. GL**	**% nac. GL**
LaLiga	503	327	65.01
Premier League	442	241	54.53
Serie A	520	277	53.27
Ligue 1	506	305	60.27
Bundesliga	468	275	58.76
Total	2439	1425	58.43

Cuadro 7: Jugadores nacidos en alguno de los cinco países.

El porcentaje de jugadores nacidos en alguno de los cinco países supera el 50 % en todas las competiciones, alcanzando más del 60 % en *LaLiga* y en la *Ligue 1*. Estas cifras son aún más significativas si tenemos en cuenta el número de nacionalidades distintas del resto de jugadores que integran los clubs de dichas competiciones (ver Cuadro 8).

Competición	**Nº nacionalidades**	**Nº no nac. GL**	**% no nac. GL**
LaLiga	50	176	34.99
Premier League	55	201	45.47
Serie A	52	243	46.73
Ligue 1	55	201	39.73
Bundesliga	57	193	41.24
Total	104	1014	41.57

Cuadro 8: Jugadores nacidos en un país distinto a cualquiera de los cinco.

Ninguna de las cinco competiciones cuenta con menos de 50 nacionalidades distintas entre los jugadores no nacidos en alguno de los cinco países de referencia. *LaLiga* es la que menos distintas presenta, 50, y la *Bundesliga* la que más, con 57. Entre las cinco competiciones existen un total de 1014 jugadores nacidos fuera de las fronteras de dichos países, proveniendo de hasta 104 países distintos. Este hecho da una idea del grado de globalización de las plantillas de estas competiciones.

A pesar de la más que demostrada globalización de estas cinco competiciones, el componente cultural del fútbol en la sociedad en cada uno de los respectivos países sigue siendo diferencial en la forma en la que tanto jugadores como entrenadores y aficionados afrontan los partidos. Para tratar de medir

el impacto que puede tener esta forma de entender el fútbol en cada uno de los países y cómo puede afectar esto al desarrollo de los partidos vamos a analizar los resultados que se dieron a lo largo de la temporada 2017-2018.

El análisis se centrará en la evolución de los resultados de los partidos, usando la condición de local y visitante para tratar de encontrar las posibles diferencias y semejanzas. Dado que las cinco ligas aglutinan a los mejores jugadores del mundo y teniendo en cuenta que están altamente globalizadas, si el componente cultural no existiese entonces cabría esperar un alto nivel de similitud en el análisis para todas ellas.

Comenzaremos nuestro análisis fijándonos en los resultados obtenidos por los equipos que actuaban como local en cada una de las competiciones:

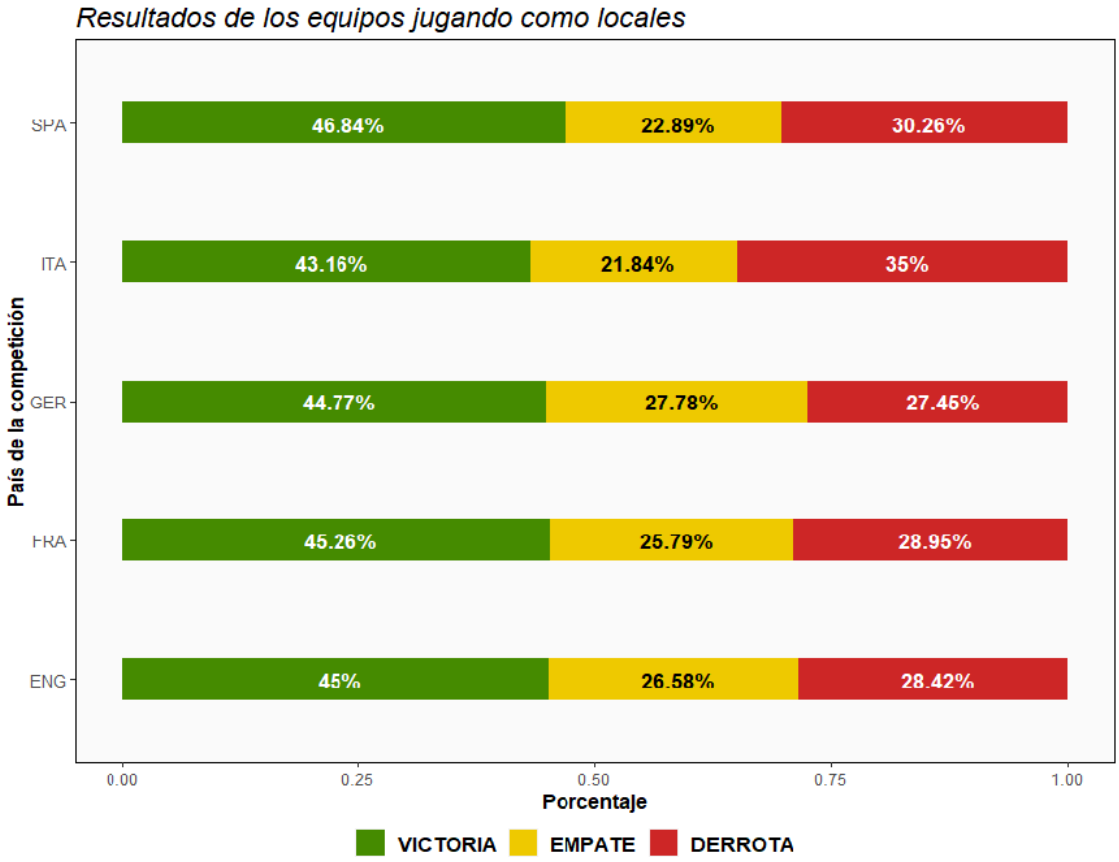

Figura 9: Resultados en función del equipo local en las 5 competiciones.

Podemos observar, en la Figura 9, como el porcentaje de victorias de los equipos locales en las cinco competiciones es bastante similar y abarca desde el *43.16 %* en la liga italiana hasta el *46.84 %* de la liga española. Entre estos dos valores se encuentran los porcentajes de victoria de los equipos locales en *Alemania, Francia e Inglaterra* comprendidos, únicamente, en un margen de 0.49 puntos porcentuales.

Estos países mantienen también un porcentaje de empates muy similar entre sí, siendo *Alemania* y *Francia* las que tienen un porcentaje más distante de las tres, con una diferencia 1.99 puntos porcentuales. Esta diferencia es algo mayor que la dada para los mismos en el caso de las victorias locales pero aún así es bastante menor a la diferencia entre el siguiente país en el que más empates se dan, *España*, y *Francia*, país con menos empates del grupo de tres mencionado. En este caso la diferencia entre *Francia* y *España* es de 2.90 puntos porcentuales, lo que sigue evidenciando la diferencia entre los dos grupos de países.

En cuanto a las victorias visitantes, y teniendo en cuenta lo comentado hasta ahora, es lógico pensar que tanto *España* como *Italia* son los países en los que más derrotas locales se producen. En efecto, esto es así, pero el porcentaje de derrotas locales en *Italia* es 4.74 puntos porcentuales superior al de *España*, cuyo porcentaje en esta ocasión es bastante más cercano al que se da en *Alemania*, *Francia* e *Inglaterra*, que siguen manteniéndose en una horquilla de 1.5 puntos porcentuales, que pasa a ser de 2.71 si sumamos a *España* a este grupo.

Hemos podido reconocer dos grupos diferenciados en cuanto a resultados se refiere: por un lado tenemos a *España* e *Italia* y por el otro a *Inglaterra*, *Francia* y *Alemania*. En el primero de estos dos grupos tenemos las competiciones en las que menos empates se producen aunque la distribución de las victorias entre los equipos que actúan como local y los que actúan como visitante no es tan similar. En el segundo de los grupos tenemos una distribución de victorias, empates y derrotas muy similar entre ellas.

Partiendo de las pequeñas diferencias encontradas debemos ir un poco más allá y analizar cómo se producen estos resultados. Para ello partiremos de la condición de local o visitante que tiene el equipo que marca el primer gol en cada uno de los encuentros y veremos cómo acaban finalizando dichos partidos.

Para los enfrentamientos que comienzan con victoria local, contamos con la representación de la Figura 10.

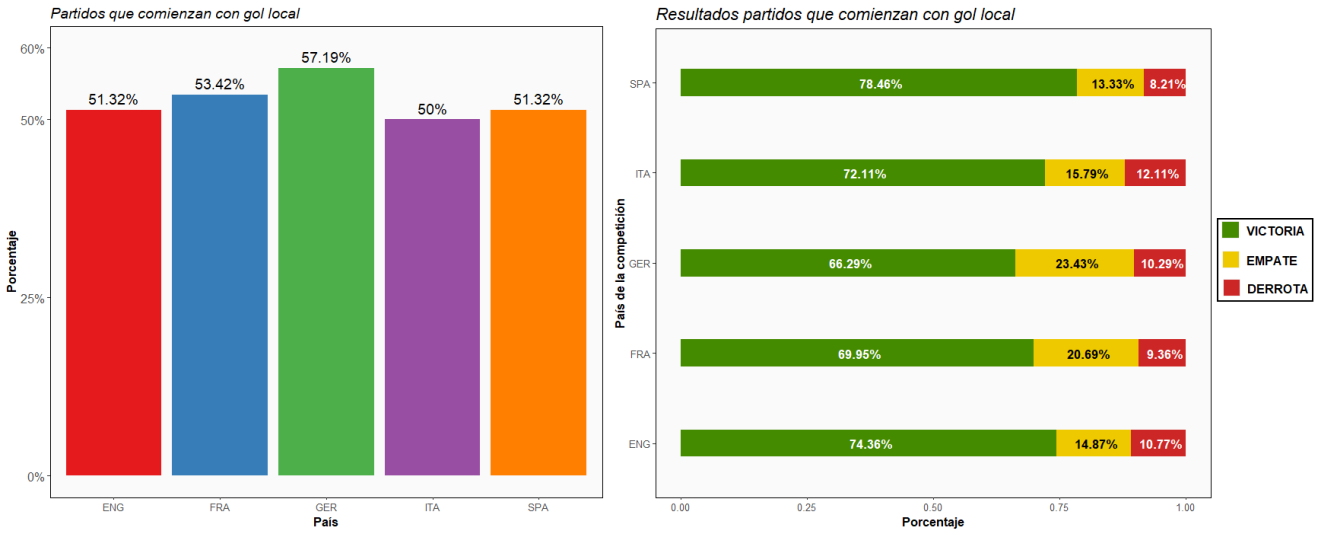

Figura 10: Resultado final de los partidos si comienza marcando el equipo local.

Vemos, en la Figura 10, como en las cinco competiciones el porcentaje de partidos en los que el primer gol es logrado por el equipo local es bastante similar, destacando por encima de todas las competiciones la *Bundesliga* alemana, con un 57.19 % de las ocasiones. En una horquilla de 3.42 puntos porcentuales se encuentran los valores para el resto de las cuatro competiciones, siendo la *Serie A* italiana la que menos porcentaje presenta con un 50 % y la *Ligue 1* francesa la que más, con un 53.42 % que representa una diferencia de 4.77 puntos porcentuales respecto a la liga alemana.

Tanto *LaLiga* española como la *Premier League* inglesa son las dos competiciones en las que más victorias locales se dan tras lograr estos el tanto inicial. En la liga española las victorias locales bajo este condicionante ascienden al 78.46 % mientras que en la competición inglesa representan el 74.36 %. La competición doméstica italiana cuenta con un 72.11 % de victorias locales mientras que la *Ligue 1* francesa acaricia el 70 % con un 69.95 %. Realmente curioso resulta de la liga alemana, que era en la que más partidos comenzaban con gol local y es a su vez, con diferencia, la liga en la que menos importancia tiene ese hecho en términos de victorias locales bajo este condicionante.

El número de partidos que acaba con victoria visitante es bastante similar en las cinco competiciones, siendo la liga italiana la que mayor porcentaje presenta con un 12.11 % y la liga española la que menos con un 8.21 %. Si comparamos los porcentajes de victorias visitantes con los de empate para cada una de las ligas se puede destacar la gran diferencia en dichos valores para la liga alemana y la liga francesa, en las que los porcentajes de los encuentros que acaban en empate son 13.14 y 10.13 puntos porcentuales superiores a los valores para el número de victorias visitantes, respectivamente. Estas diferencias tan amplias se hacen aún más evidentes si las comparamos con las otras tres competiciones, que no sobrepasan los 6 puntos porcentuales de diferencia.

Por tanto, hemos visto que aunque en las cinco competiciones el número de partidos que inician con gol local es similar, su impacto en el resultado final no es el mismo en cada una de ellas. En *LaLiga*

española y en la *Premier League* inglesa el primer gol local se suele traducir en un mayor número de ocasiones en el triunfo del equipo anfitrión, mientras que en la *Bundesliga* alemana encontramos el menor número de victorias locales bajo este condicionante. La *Serie A* italiana es la competición en la que más victorias visitantes se dan tras un primer gol local. Por último, la *Ligue 1* francesa y la *Bundesliga* son las competiciones en las que más empates se dan cuando el equipo local comienza marcando, siendo la diferencia en ambas entre el porcentaje de empates y el porcentaje de victorias visitantes mucho más amplio que en cualquiera de las otras tres competiciones.

Una vez establecidas las diferencias y similitudes en los resultados de los encuentros que iniciaban con tanto de la escuadra local, es hora de plantear el escenario contrario y ver que ocurre cuando comienza ganando el equipo visitante (ver Figura 11).

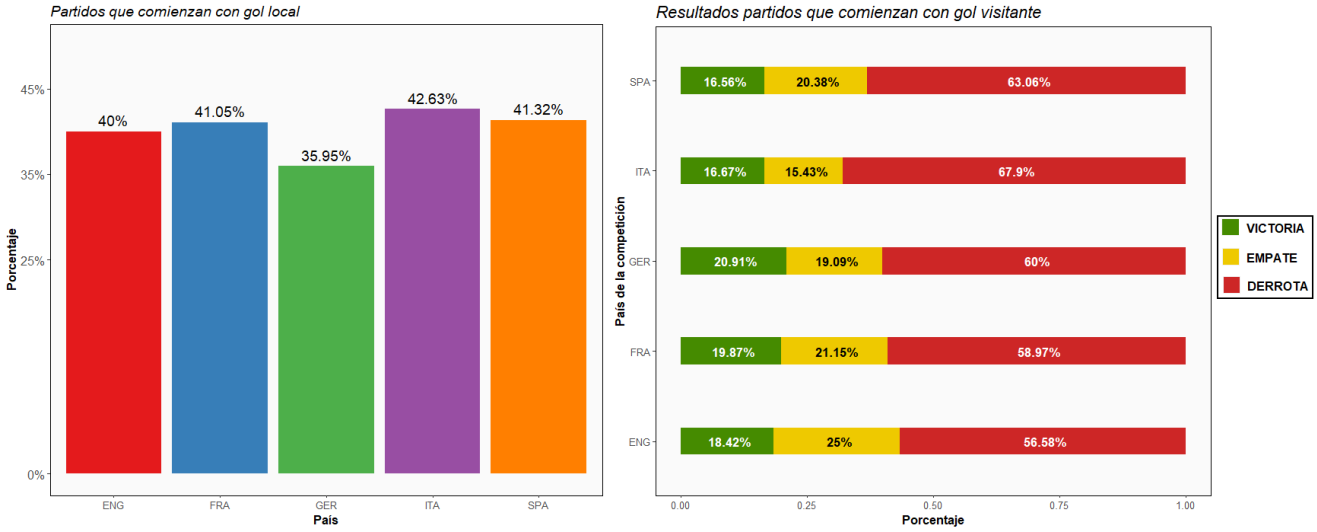

Figura 11: Resultado final de los partidos si comienza marcando el equipo visitante.

La *Bundesliga* alemana es la competición en la que más victorias visitantes se dan cuando el equipo local comienza perdiendo, ocurriendo en el 67.9 % de los partidos en los que esto ocurre. Le sigue *LaLiga* española a una diferencia de 6.84 puntos porcentuales, casi la misma diferencia que mantiene respecto a la *Premier League* que es la competición en la que menos victorias visitantes se dan cuando comienzan adelantándose en el marcador, con un porcentaje del 56.58 %.

Si nos fijamos en los porcentajes de los partidos que acaban empate y los que acaban en victoria local vemos que la mayor diferencia entre ambos valores se da en la competición inglesa, con 6.58 puntos porcentuales más en el caso de empate. Esta misma comparación hecha en la Figura 10 había arrojado diferencias significativas para la competición francesa y la alemana, pero en este caso vemos que las diferencias son mínimas. Respecto a la *Serie A* italiana y *LaLiga* española tenemos unos porcentajes prácticamente idénticos para los partidos que finalizan con victoria local, y una diferencia de casi 4.5 puntos porcentuales para el número de empates.

Para los enfrentamientos en los que comienza marcando el equipo visitante vemos una cierta similitud entre los resultados finales para la competición alemana y la francesa por un lado, y la italiana y la española por otro, aunque en menor medida.

En los dos escenarios planteados hasta ahora hemos podido observar como las mayores diferencias entre las distintas ligas se daban en los partidos que finalizaban en empate, por lo que a continuación vamos a tratar de entender cómo se llega a estos resultados.

Para ello, contamos con la representación de la Figura 12.

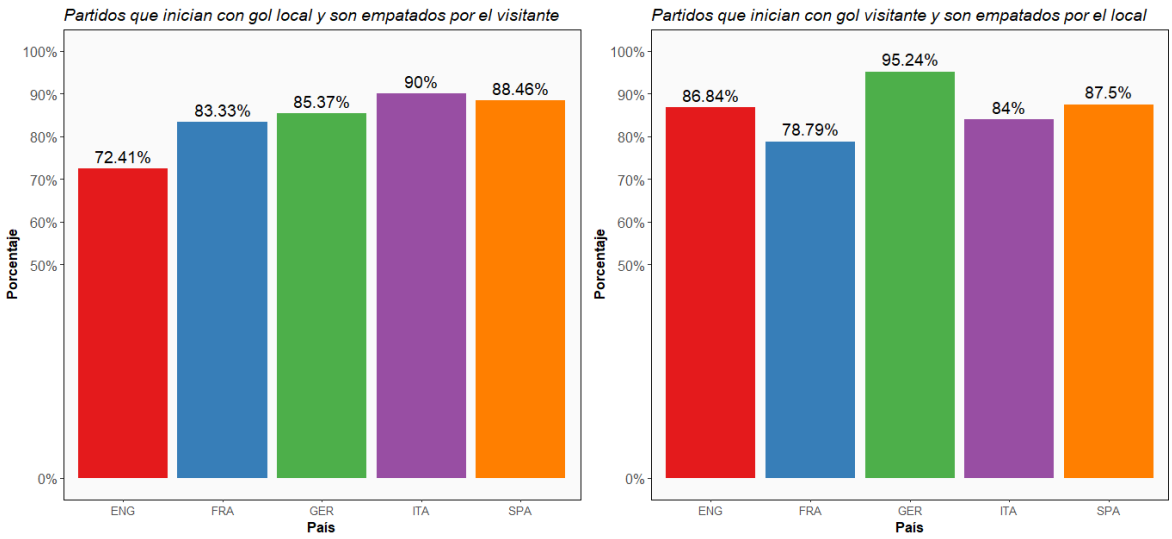

Figura 12: Quienes marcan el último gol en los partidos que finalizan en empate.

En el gráfico de la derecha, de la Figura 12, tenemos representado el porcentaje de los partidos que tras un gol local inicial acaban empate tras un último gol visitante. En el gráfico de la izquierda, tenemos la situación contraria, es decir, el porcentaje de partidos que tras un gol inicial del equipo visitante acaban empate tras un último gol del equipo local.

La *Premier League* inglesa es la competición que más destaca en el primero de los gráficos, de la Figura 12, ya que tiene un porcentaje bastante bajo en comparación a las otras cuatro ligas representadas. El 72.41 % de los partidos que finalizan en empate tras lograr el equipo local el tanto inicial del partido son consecuencia de un último gol del equipo visitante, lo que visto desde el punto de vista contrario significa que en el 28.59 % de estos partidos el equipo visitante logra dar la vuelta al resultado para finalmente ser empatados por el equipo local. En contraposición tenemos la *Serie A* italiana, que presenta un 90 % de partidos empatados por el equipo visitante tras adelantarse el equipo local. Entre estos dos valores se encuentras las otras tres ligas domésticas objeto de estudio, con valores más cercanos al porcentaje de la liga italiana que a la inglesa.

Respecto al segundo gráfico, de la Figura 12, la *Ligue 1* francesa presenta el porcentaje más ba-

jo con un 78.79 % de empates provocados por el equipo local tras lograr el equipo visitante el gol inicial del partido. Esto quiere decir que en el 21.21 % de los partidos que comienzan con gol inicial visitante y acaban en empate el equipo local consigue remontar el partido para finalmente ser empatados por los visitantes. El caso opuesto se da en la *Bundesliga* alemana, en el que el 95 % de los empates que se producen tras un gol inicial visitante se producen por un gol final del equipo local.

En la Figura 12 hemos podido ver cómo se dan los empates en función del equipo que comienza marcando, lo que nos ayuda a tener una visión algo más amplia sobre cómo de disputados y abiertos son los encuentros en cada una de las competiciones. Cuándo es el equipo local el que comienza marcando hemos visto que es más factible una remontada, posteriormente frustrada, en la liga inglesa que en cualquiera de las otras cuatro competiciones lo que puede llevarnos a la interpretación de que la *Premier League* es la liga en la que se dan los partidos con más alternativas en el marcador entre los equipos contrincantes, mientras que en la *Serie A* ocurre todo lo contrario.

En el caso de que el partido acabe en empate, tras comenzar con gol visitante, la *Bundesliga* es la competición en la que más empates son consecuencia de un último gol local, con un 95.24 %, por lo que junto a lo visto en la Figura 11 podemos pensar que en dicha competición los partidos que comienzan con un primer gol visitante terminan convirtiéndose en un asedio del equipo local sobre el arco rival. El caso opuesto se da en la *Ligue 1*, ya que en uno de cada cinco partidos que finalizan en empate tras un gol inicial visitante se llega a la igualdad en el marcador por culpa de un gol visitante, lo que puede llevarnos a pensar que en la liga francesa bajo este condicionante se dan partidos más abiertos, en comparación a las otras cuatro competiciones, en el que cualquier equipo puede llevarse la victoria.

Hemos sido capaces de detectar múltiples diferencias entre las competiciones, aunque es cierto que al comienzo identificamos dos grupos de ligas, ver Figura 9. El primero de ellos formado por *LaLiga* y la *Serie A* y el segundo por la *Bundesliga*, la *Premier League* y la *Ligue 1*. El análisis a través de los distintos escenarios planteados nos ha llevado a distinguir diferencias significativas entre ligas de los mismos grupos iniciales e incluso semejanzas entre competiciones de grupos distintos en un principio.

Por tanto, podemos pensar que a pesar de la *globalización* de estilos y tácticas en el ámbito futbolístico, el componente cultural sigue siendo determinante ya que continúa marcando las diferencias entres los distintos países. Este hecho, ligado a la gran cantidad de jugadores de estas ligas que disputaron el *Mundial de Rusia*, hace de *LaLiga*, *Premier League*, *Serie A*, *Ligue 1* y *Bundesliga* la mejor representación posible para la creación de un modelo cómo el que se plantea en este trabajo.

5. Conjunto de datos para el modelo

En este capítulo, en primer lugar, se van a exponer los criterios que debe reunir un tiro para que sea considerado por nuestro modelo de goles esperados. En segundo lugar, se va a realizar un análisis descriptivo de las variables que van a ser introducidas en el modelo.

5.1. Criterios de selección

Cómo vimos en el Capítulo 4 vamos a emplear los datos de *LaLiga*, *Premier League*, *Serie A*, *Ligue 1* y *Bundesliga*, de la temporada 2017-2018, para construir un modelo de goles esperados para evaluar las ocasiones generadas en los partidos del *Mundial de Rusia* de 2018.

Concretamente nos interesan las características propias de cada uno de los tiros realizados en dichas competiciones, pero antes debemos definir que tiros son útiles y cuales no para el objetivo que tenemos. Los criterios que debe cumplir un tiro para ser considerado útil son los siguientes:

- *Ser un tiro intencionado*: las acciones catalogadas como tiro pero que son producto de un error de uno de los defensores, no serán consideradas. Únicamente usaremos los tiros realizados por el equipo atacante.
- *No ser interceptado*: un tiro será considerado únicamente si su desenlace es uno de los tres siguientes: gol, parada del portero o finalización de jugada. Todos aquellos tiros que son desviados y/o bloqueados por un defensor o atacante no se tendrán en cuenta.
- *Darse en una jugada a balón corrido*: no se considerarán los tiros a puerta efectuados desde el punto de penalti, ni los lanzamientos de falta ni los tiros desde el banderín de córner, los conocidos como goles olímpicos.

El primer y segundo criterio son inherentes a la naturaleza de un modelo de goles esperados. El primero, restringe los tiros a considerar a aquellos que han sido realizados por el equipo atacante. Esto es así porque la métrica de goles esperados tiene como objetivo evaluar la producción ofensiva de los equipos, por lo que fallos del equipo rival, como los remates en su propia portería, no definen el potencial ofensivo de un equipo, y por tanto, estas situaciones no deben ser consideradas. El segundo, establece que únicamente se consideran los tiros que han finalizado de forma natural, sin que ningún jugador, salvo el portero contrario, intervenga en la trayectoria del disparo. Los tiros que son bloqueados o interceptados durante su trayectoria ya se sabe que no pueden finalizar en gol, por lo que no se deben tener en cuenta.

El tercero de los criterios ha sido incluido debido a las circunstancias propias del conjunto de datos disponible. En realidad, la métrica de goles esperados sí evalúa, por ejemplo, la probabilidad de conversión de un penalti, pero al ser nuestro conjunto de datos tan limitado, se ha descartado el cálculo para estas situaciones.

El número de tiros por competición, tanto considerando los condicionantes como sin considerarlos, se muestran en el Cuadro 9.

Competición	**Total tiros**	**Tiros a considerar**	**% considerados**
LaLiga	8545	6231	72.92
Premier League	8881	6179	69.58
Serie A	9347	6742	72.13
Ligue 1	8977	6438	71.71
Bundesliga	7290	5321	71.75
World Cup	1573	1036	65.86
Total	44613	31947	71.61

Cuadro 9: Número de tiros por competición

En total tenemos información de 44613 tiros, pero únicamente nos son de utilidad el 71.61 % de estos, es decir, 31947 lanzamientos a puerta. Como veremos en el Capítulo 6.1.3, los 30911 correspondientes a las competiciones domésticas serán los que emplearemos para generar el modelo mientras que los 1036 de la *Copa Mundial* no serán empleados para entrenar el modelo ya que son los tiros que queremos evaluar.

Al considerar únicamente los tiros que son a balón corrido nuestro conjunto de datos cuenta con menos goles que los que realmente se lograron en las seis competiciones. El desglose por competición es el siguiente:

Competición	**Total goles**	**Goles considerados**	**% considerados**
LaLiga	1024	884	86.33
Premier League	1018	913	89.69
Serie A	1017	853	83.88
Ligue 1	1033	873	84.51
Bundesliga	855	747	87.37
World Cup	169	128	75.74
Total	5116	4398	85.97

Cuadro 10: Número de goles por competición

Los goles que no son contemplados se debe a que o bien fueron en propia puerta o bien se dieron a balón parado. La *Premier League* es la competición que aporta un mayor número de goles a nuestro

conjunto de datos, mientras que la *Serie A* es de la que menos goles tenemos, en porcentaje, de las cinco grandes ligas.

5.2. Variables consideradas

Tras definir los tiros que son de nuestro interés y conocer su desglose por competición, así como los goles contenidos en dichos tiros, es hora de definir las características propias de cada tiro que vamos a considerar en el modelo. En total contamos con 15 variables, 14 de ellas explicativas y una última que es la respuesta. Si dividimos dichas variables por su naturaleza, tenemos: 4 variables categóricas y 11 variables numéricas.

5.2.1. Variables categóricas

De las cuatro variables categóricas con las que contamos, tres de ellas son variables explicativas y la cuarta es la variable respuesta.

- **gol**:
  Es la variable respuesta, y se codifica en $\{0\text{-}1\}$: en caso de que el tiro en cuestión sea fallido lo codificamos con *0*, mientras que si es un intento exitoso se codifica con *1*. En total, ver Cuadro 9, contamos con 31947 tiros para construir nuestro modelo de los cuales, como vimos en el Cuadro 10, 4398 son goles. Esto significa que el *86.23 %* de los intentos son no exitosos frente al *13.77 %* que sí lo son.

- **golpeo**:
  Esta variable recoge la zona del cuerpo con la que el jugador ha realizado el lanzamiento, teniéndose tres posibilidades: *dcha* si lo realiza con su pierna derecha, *izda* si es con su pierna izquierda y *cab/cuerpo* si es con la cabeza o cualquier otra parte del cuerpo distinta a las anteriores. Para la generación de esta variable se ha recurrido a las columnas *tags* asociadas al propio tiro.

Golpeo/Gol	**% no gol**	**% gol**	**% acum.**
Derecha	42.01	6.66	48.67
Izquierda	26.78	4.44	31.22
Cab/Cuerpo	17.45	2.66	20.1
Total	86.24	13.76	100.0

Cuadro 11: Tabla de contingencia para la variable *golpeo*

Si nos fijamos en el Cuadro 11 vemos que prácticamente la mitad de los tiros se realizaron con la pierna derecha, suponiendo el 48.67 % del total. Con el perfil zurdo se realizaron 3 de cada 10 lanzamientos, 31.22 %, mientras que con la cabeza y/u otras partes del cuerpo el 20.1 % restante.

Golpeo/Gol	**% no gol**	**% gol**
Derecha	48.71	48.40
Izquierda	31.05	32.27
Cab/Cuerpo	20.24	19.33
Total	100.0	100.0

Cuadro 12: Tabla de frecuencias para la variable *golpeo*

De la misma manera, el número de goles conseguidos con cada uno de los tipos de golpeo es similar al porcentaje de tiros sin éxito realizados con cada uno de ellos, tal y cómo se observa en el Cuadro 12. Esto indica que no existe una alta asociación entre esta variable y la respuesta. El test chi-cuadrado sobre esta variable nos lleva a la misma conclusión, con un p-valor asociado de 0.1755 que invita a no rechazar la hipótesis nula de no independencia.

A pesar de ello, mantendremos la variable dentro del modelo ya que, basándonos en otros modelos, es bien sabido que realizar un tiro con una u otra pierna puede ser de gran importancia en función de la distancia a la portería, el ángulo, la zona del campo y otras variables que también serán incluidas en el modelo y que veremos más adelante.

- **golpeoHabil**:
  Mediante esta variable indicamos si el jugador que realiza el tiro lo hace ayudándose de su pierna dominante o si por el contrario lo realiza con la pierna menos hábil u otra parte del cuerpo. Por tanto, para cada tiro tenemos dos posibles categorías atendiendo a este criterio: *habil* si lo realiza con su pierna dominante, *no-habil* si lo realiza con su pierna menos hábil o si golpea el balón con la cabeza u otra parte del cuerpo. Para la creación de esta variable, se ha empleado la variable *golpeo* y la información contenida en *players.csv* sobre el jugador que lo realiza.

Hábil/Gol	**% no gol**	**% gol**	**% acum.**
Hábil	53.40	8.30	61.70
No hábil	32.83	5.47	38.30
Total	86.23	13.77	100.0

Cuadro 13: Tabla de contingencia para la variable *golpeoHabil*

Hábil/Gol	**% no gol**	**% gol**
Hábil	61.93	60.28
No hábil	38.07	39.72
Total	100.0	100.0

Cuadro 14: Tabla de frecuencias para la variable *golpeoHabil*

Como podemos ver en el Cuadro 13 el 61.70 % de los tiros efectuados se realizan con la pierna hábil, mientras que el 38.30 % restante se efectúan con la pierna menos hábil del jugador o con alguna otra parte del cuerpo. De nuevo, ver Cuadro 14,no se observan diferencias para cada categoría en función de la variable respuesta. Sin embargo, el test chi-cuadrado asociado a esta variable si las detecta, arrojando un p-valor de 0.03322 que permite rechazar la hipótesis nula de de no independencia.

Para la creación de esta variable se consideró la opción de incluir un tercer nivel que distinguiese de entre los tiros categorizados como "No hábil" aquellos que realmente se realizasen con la cabeza u otra parte distinta del cuerpo. Finalmente, se descartó esta idea ya que esta información ya se incluye en el modelo por medio de la variable *golpeo*.

- **accionPrevia**:
  Tal y como su nombre indica, refleja qué tipo de acción tuvo lugar justo antes del tiro. En total existen hasta 11 valores distintos, y son: *Acc. Individual*, *Acc. Portero Contrario*, *Balón suelto*, *Centro*, *Pase*, *Pase alto*, *Pase inteligente*, *Reanudación juego*, *Robo*, *Tiro previo* y *Otros*. Su creación se realiza a partir de la información del evento que antecede el tiro y en función de la naturaleza de ese evento se incluye en alguna de las 11 categorías. El proceso de creación de esta variable se describe en el Anexo C.1.

accionPrevia/Gol	**% no gol**	**% gol**	**% acum.**
Acc. Individual	18.55	2.41	20.96
Acc. Portero Contrario	1.20	0.81	2.01
Balón suelto	8.53	1.71	10.24
Centro	25.33	4.42	29.75
Otros	0.07	0.01	0.08
Pase	19.33	2.28	21.61
Pase alto	2.05	0.36	2.41
Pase inteligente	3.74	1.07	4.81
Reanudación juego	0.65	0.03	0.68
Robo	6.40	0.47	6.87
Tiro previo	0.38	0.20	0.58
Total	86.23	13.77	100.0

Cuadro 15: Tabla de contingencia para *accionPrevia*

La mayor parte de los tiros se categorizan bajo cuatro tipo de acciones previas: *Centro* que representa casi el 30 % del total, *Pase* y *Acc. Individual* superan ligeramente el 20 % y *Balón suelto* que representa el 10 % (ver Cuadro 15). Estas cuatro categorías representan el 82.56 % del total lo que provoca que el resto se presenten en porcentajes muy bajos, en algunos caso prácticamente nulos como la categoría *Otros*. Aún así, se ha optado por mantener todas y cada una de ellas ya que representan situaciones reales de juego muy diferenciadas por lo que unir varias de estas categorías podría dar lugar a peores resultados en el construcción del modelo.

5.2.2. Variables numéricas

Hasta 11 de las 15 variables con las que contamos son de tipo numérico. Los nombres de dichas variables son: *resultado*, *goles_equipo*, *posición*, *distancia*, *ángulo*, *segundos90*, *ultTiro*, *ultGol*, *tiempoJugada*, *distanciaJugada* y *velocidadJugada*. A continuación, se van a exponer todas y cada una de ellas.

- **resultado**:
  Esta variable mide la diferencia de goles en el partido entre los equipos en el instante previo al golpeo. Se mide siempre desde la perspectiva del equipo que realiza el tiro, midiendo de esta forma cuantos goles de ventaja o desventaja tiene ese equipo en dicho momento. Si en un partido con resultado 2 a 1, el equipo que va ganando realiza un disparo, esta variable toma el valor $2-1=1$; si por el contrario, lo realiza el equipo que va perdiendo el valor será $1-2=-1$. Con la adición de esta variable se busca medir la desigualdad entre los equipos contrincantes,

así cómo el factor psicológico de ir por delante o por detrás en el marcador.

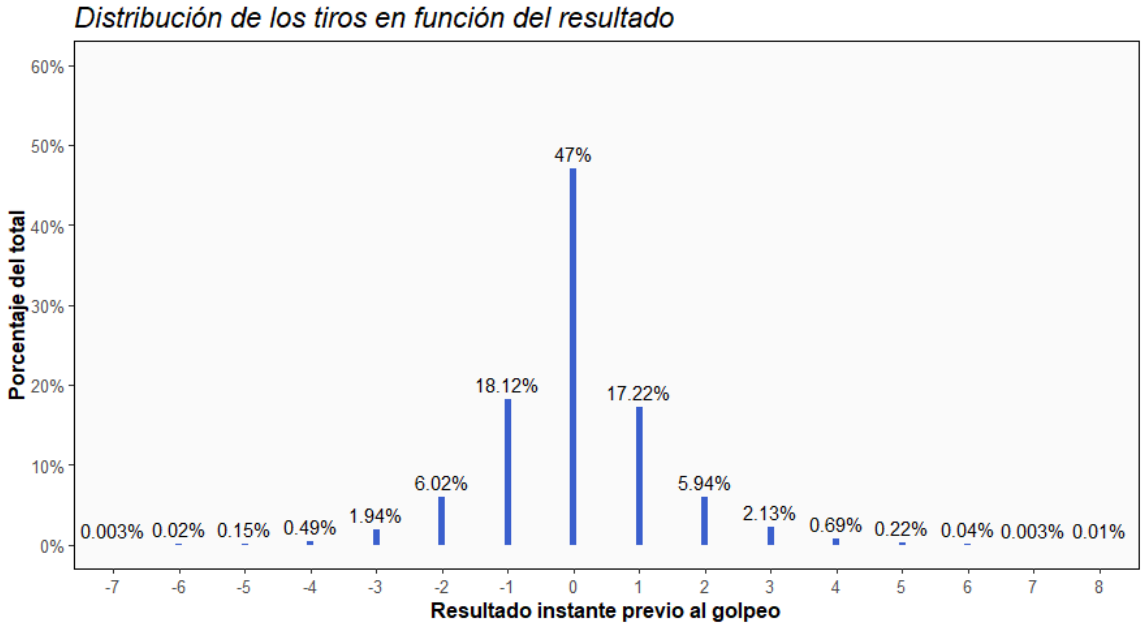

Figura 13: Distribución de los tiros en función del resultado.

El rango de valores para esta variable abarca desde el -7 hasta 8. Estos valores extremos son apenas distinguibles en el gráfico ya que para el extremo inferior únicamente contamos con dos casos y para el superior solo con uno. Lógicamente la mayor parte de los tiros se realizan bajo un valor de *resultado* igual a cero ya que los partidos comienzan en una situación inicial de empate. En el intervalo [-3,3] se aglomera prácticamente la totalidad de los tiros quedando un número residual de lanzamientos en el resto de posibles resultados fuera de dicho intervalo, lo que significa que las situaciones en las que un equipo va ganando, o perdiendo, por más de tres goles son poco frecuentes.

- **goles_equipo**:
  Esta variable está muy relacionada con *resultado* ya que indica el número de goles conseguidos por el equipo que efectúa un lanzamiento en el instante previo a su realización. Su inclusión en el modelo permitirá medir la capacidad goleadora del equipo en cuestión en dicho partido, así como conocer con más exactitud el escenario real del partido en cuanto a resultado se refiere.

  A la hora de evaluar un tiro no es lo mismo que un equipo vaya ganando por un único gol en un partido en el que solo se ha marcado ese gol a que lo haga en un partido en el que se han marcado cinco goles. El primer escenario puede indicar un alto nivel defensivo de los equipos contrincantes, mientras que en el segundo escenario lo más plausible es que los atacantes se

estén imponiendo a los defensores en ambos equipos.

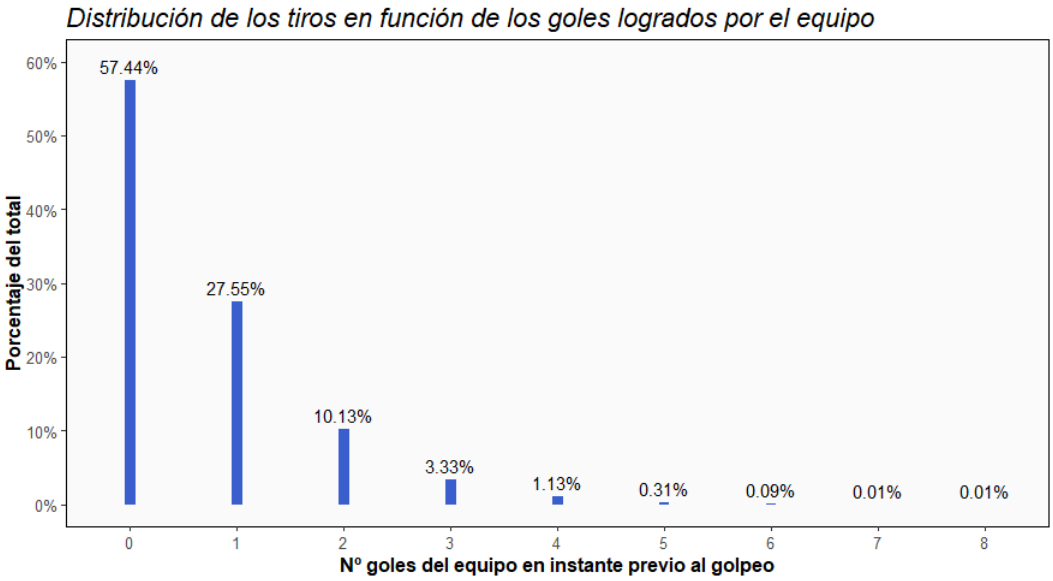

Figura 14: Distribución de los tiros en función de los goles ya logrados por el equipo que lanza.

La distribución que podemos apreciar en la Figura 14 guarda una gran relación con la vista en el Cuadro 13 para *resultado*, y es que, como habíamos adelantado, ambas variables están bastante ligadas dándonos información complementaria que nos permite reconstruir el marcador exacto en el instante de golpeo.

- **posición**:
  Tal y como vimos en el Capítulo 4.1.1, *Sam Green* [21] concluyó que la *posición* del campo desde la que se realiza el tiro es de vital importancia a la hora de determinar el éxito de un lanzamiento. Para introducir la posición del tiro en el modelo empleamos las variables **xf** e **yf** que representan el par de coordenadas *(x,y)* desde el que se efectúa.

Figura 15: Posiciones desde las que se realizaron los tiros que acabaron en gol.

En la Figura 15 se aprecia que la mayor parte de los goles se consiguen desde dentro del área, especialmente en las zonas más centradas respecto a la portería. Los goles conseguidos desde fuera del área también se concentran en la zona central del campo, siendo muy pocos los goles conseguidos desde posiciones escoradas. Si representamos las posiciones de los tiros en un mapa de calor, como en la Figura 16, podemos apreciarlo mejor.

Figura 16: Mapa de calor para las posiciones de los tiros que finalizaron en gol.

Los colores más rojizos, que se corresponden con un mayor nivel de ocurrencia, se encuentran en torno al punto de penalti, lo que quiere decir que desde esa posición se lograron un mayor número de goles que desde cualquier otra parte del campo. Según el lanzamiento se va alejando

de esa posición, ya sea hacia los lados o hacia atrás, el número de ocurrencias va disminuyendo, hasta tal punto que zonas que cuentan con algún gol, ver Figura 15, no figuran en el mapa, indicio de que es poco habitual.

- **distancia**:
  La *distancia* se mide desde la *posición* de disparo al centro de la portería. Su inclusión en el modelo es de vital importancia como se indicó en el trabajo de *J. Ensum*, *R. Pollard* y *S. Taylor* [18]. En la Figura 17 podemos ver, sobre nuestro conjunto de datos, esta idea.

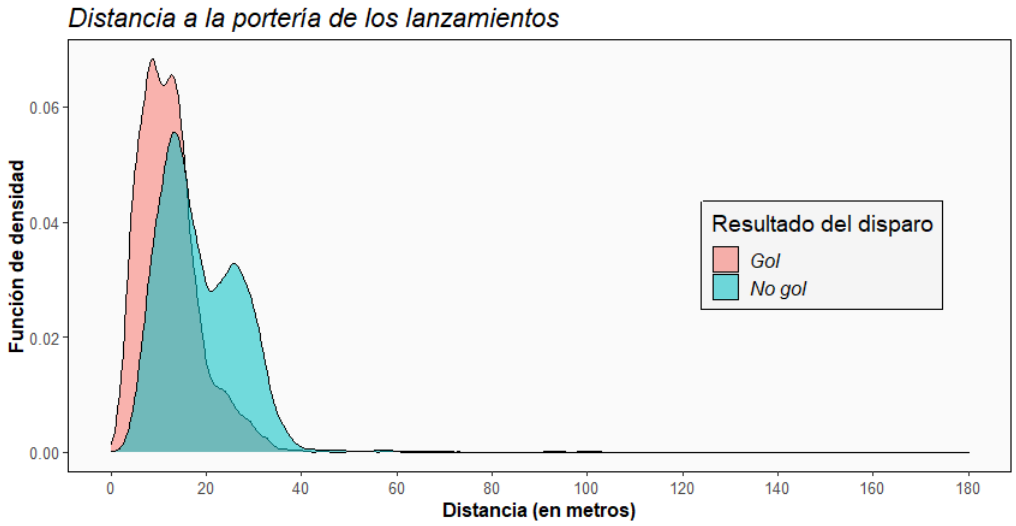

Figura 17: Función de densidad estimada para la distancia de los tiros efectuados según resulten en gol o no.

En el rango de 0 a 18 metros hay una concentración mucho mayor de tiros que finalizan en gol que de tiros sin éxito. Esta diferencia es tal, que en ese rango se concentran el 85.61 % de lo goles, frente al 53.82 % de tiros que no resultan en gol. Además, la proporción de tiros efectuados por cada tiro que acaba en gol en distancias menores a 18 metros es de un gol por cada cinco tiros. Para distancias superiores a los 18 metros, esta proporción se va hasta los 21 tiros por cada gol. Si observamos las medias y medianas de ambas situaciones también percibimos una gran diferencia. En el caso de los tiros que finalizan en gol la media es 12.18 metros y la mediana 11.42 metros. Para el escenario contrario, estos valores ascienden a 18.74 metros y 17.10 metros.

- **ángulo**:
  La variable *ángulo* hace referencia a la amplitud de tiro con la que cuenta el atacante desde la ubicación en la que lanza a puerta. Para hallar este valor se tiene en cuenta tanto la posición del atacante como las coordenadas en las que se encuentran los palos y el centro de la portería. En el Anexo C.3 se explica con mayor detalle este concepto.

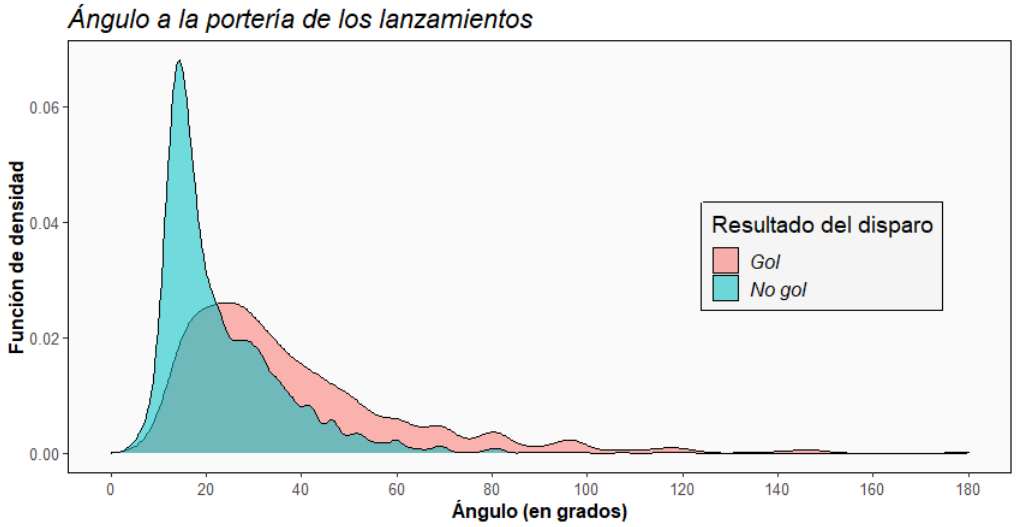

Figura 18: Función de densidad estimada para la variable ángulo según el disparo resulte en gol o no.

La relación entre la variable *ángulo* y el éxito de conversión queda clara en la Figura 18. El 63.03 % de los tiros que no finalizan en gol se dan bajo un ángulo igual o menor a los 23^{o}. Por el contrario, bajo esa misma situación se producen únicamente el 28.49 % de los goles. A partir de esos 23^{o} existe una densidad notablemente mayor de tiros que finalizan en gol que de tiros no exitosos. Si calculamos la proporción de goles por tiros efectuados bajo un ángulo menor a 23^{o} obtenemos que por cada gol se han realizado 14.85 tiros. Esta misma proporción calculada para ángulos mayores a dicho umbral es de 4.24 tiros por cada gol.

- **segundos90**:
  Esta variable mide, en segundos, el tiempo transcurrido desde el inicio del partido hasta el instante del golpeo. En los partidos que se disputan bajo el formato regular abarca desde el 0.0 hasta el 5400.0, mientras que si se disputa también el tiempo de prórroga se prolonga desde el 0.0 hasta el 7200.0.

  Los tiempos de descuento de la primera y segunda parte se acumulan en el valor 2700.0 y 5400.0, respectivamente, que equivalen al minuto 45 y 90 de partido. Esto se hace así para evitar confundir el minuto 1 de la segunda parte con el minuto 46 de la primera. En el caso de haber prórroga, los tiempos de añadido de dichas partes también se acumulan en el minuto final de cada una de ellas, es decir, el añadido de su primera parte se registra como el instante 6300.0 mientras que para el descuento de la segunda parte se considera el instante 7200.0.

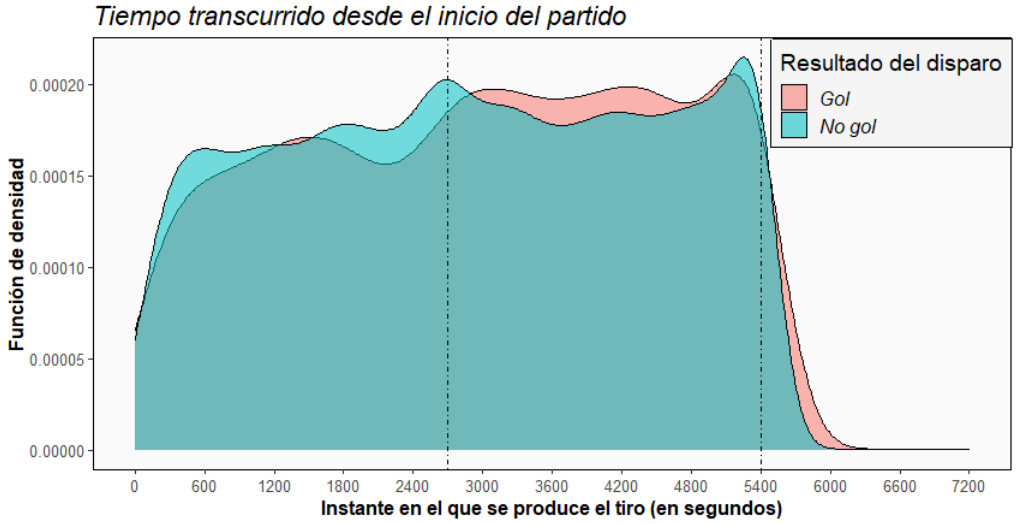

Figura 19: Función de densidad estimada para la variable *segundos90* según el disparo resulte en gol o no.

Las dos funciones de densidad representadas en la Figura 19 son bastante similares. Durante los instantes comprendidos entre los segundos 2900 y 5200, aproximadamente, la densidad de goles es superior a la de los tiros fallidos. Esta diferencia no es elevada, ya que en ese intervalo se producen el 44.57 % de los goles frente al 42.51 % de tiros sin éxito.

Existe un segundo intervalo en el que la concentración de goles es mayor que la de tiros no convertidos, y se da entre los segundos 5600 y 6300 que corresponden al primer tiempo de la prórroga. Como el número de muestras en el conjunto para ese intervalo es muy bajo, no le daremos mayor relevancia en el análisis a dicha situación.

- **ultTiro**:
  La variable *ultTiro* mide el tiempo que ha transcurrido desde el último tiro que realizó el mismo equipo que protagoniza el lanzamiento a considerar. En el caso de ser el primer tiro que se realiza desde el comienzo, o descanso, del partido, se asigna el valor del tiempo transcurrido desde entonces.

  Para el cálculo de esta variable se han utilizado también la información de aquellos tiros que no cumplen con los requisitos establecidos al comienzo de este Capítulo 5 como para ser considerados por nuestro modelo de goles esperados, ya que aportan información de gran valor para aquellos tiros que sí son incluidos. Por ejemplo, los lanzamientos que son interceptados por un jugador rival no forman parte del conjunto de tiros de interés. Sin embargo, es posible que tras ese primer intento no considerado se produzca un segundo lanzamiento que sí sea de nuestro interés. En ese caso, el tiempo transcurrido desde el último tiro debe medirse usando la información del tiro interceptado, y no la del último tiro válido.

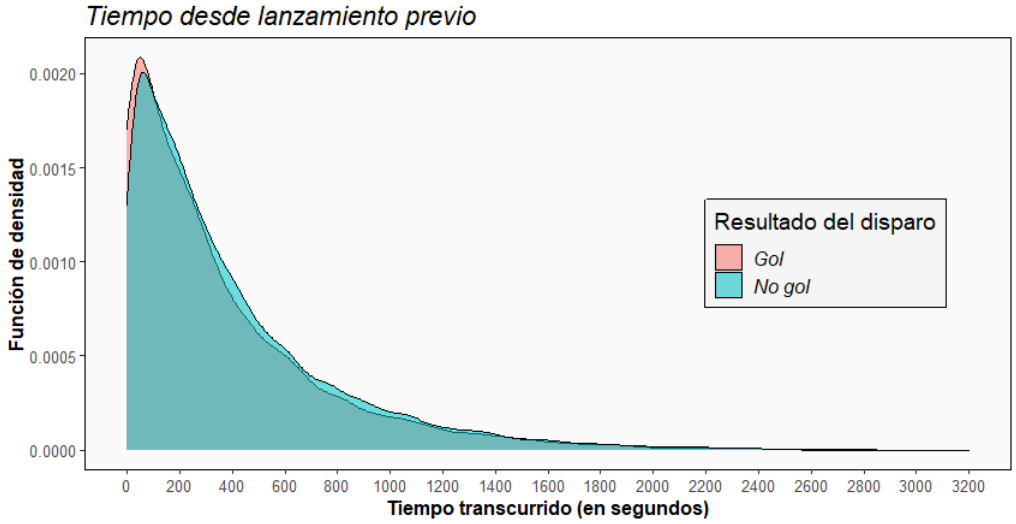

Figura 20: Función de densidad estimada para el tiempo transcurrido entre tiros consecutivos de un mismo equipo, según resulten en gol o no.

Como podemos ver en la Figura 20 las dos funciones de densidad representadas son prácticamente idénticas. En el intervalo [0,200] se producen el 40.91 % del total de los tiros, habiendo una mayor densidad de los tiros que finalizan en gol que de los tiros sin éxito. Concretamente, en dicho intervalo se producen el 32.38 % de los goles frente al 28.99 % de los tiros fallidos. Tanto el alto grado de concentración de tiros en dicho intervalo, como la diferencia presentada entre ambas situaciones, puede deberse a dos motivos.

El primero de ellos son los lanzamientos realizados tras un rebote o la construcción de una jugada tras un tiro previo, lo que facilita que el equipo defensor se encuentre en situación desfavorable y se incremente la facilidad de conversión. El segundo se trata de las situaciones de partido en las que un rival es claramente superior y asedia de forma constante a su rival, encadenando una gran cantidad de ocasiones en un breve periodo de tiempo.

- **ultGol**:
  Mide el tiempo transcurrido desde que se produjo el último gol del partido hasta el instante del golpeo. Al igual que ocurría con la variable *ultTiro*, todos los tiros realizados antes de que se produzca el primer gol consideran como valor para esta variable el tiempo transcurrido desde el inicio del partido o de la segunda parte. Los goles logrados en el primer periodo no son considerados para el cálculo de los tiros de la segunda mitad.

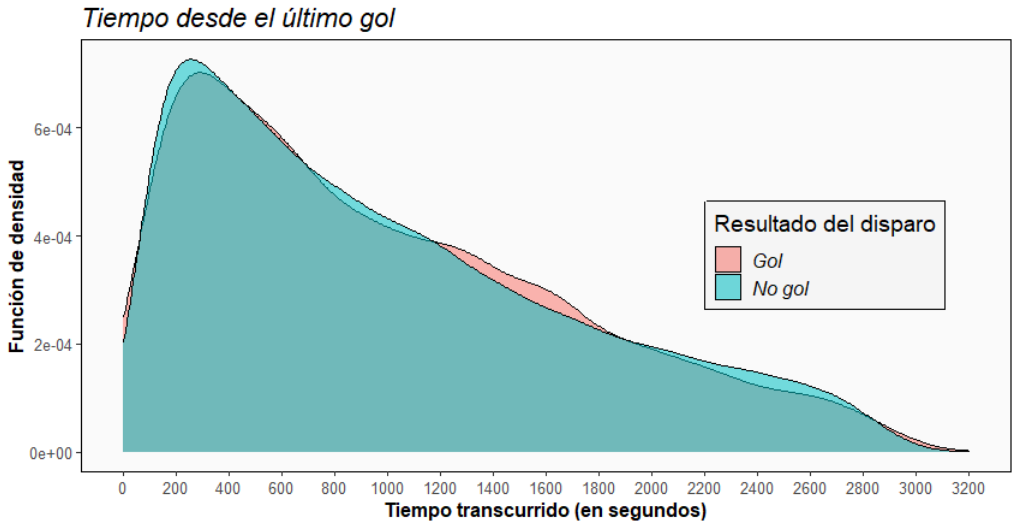

Figura 21: Función de densidad estimada para el tiempo transcurrido desde el último gol según el resultado del disparo.

Para esta variable, como se muestra en la Figura 21, tenemos dos funciones de densidad muy similares. El único intervalo en el que la densidad de los tiros que finalizan en gol es mayor que la de los tiros sin éxito es el [1200,1800]. Entre esos instantes de tiempo se concentran el 24.10 % de los goles y el 22.47 % de los tiros fallidos. A pesar de ser una diferencia pequeña, es la más grande existente entre ambos escenarios.

- **tiempoJugada**:
  Mide el tiempo que ha transcurrido desde el comienzo de la jugada hasta que se realiza el tiro. Con su introducción en el modelo se busca medir el grado de elaboración de la jugada, ya que una jugada que dure 5.0 segundos tendrá menos pases y elaboración que otra que dure, por ejemplo, 30.0 segundos.

  Aquellos tiros que no son fruto de una jugada sino que son consecuencia de un rebote o balón suelto, toman el valor 0.0. Para el resto de casos, es necesario localizar el evento que da inicio a la jugada para poder medir el tiempo transcurrido hasta el evento del disparo. Con ese fin, se han definido una serie de patrones para detectar el fin de una jugada y comienzo de otra. Estos patrones se explican con detenimiento en el Anexo C.2.

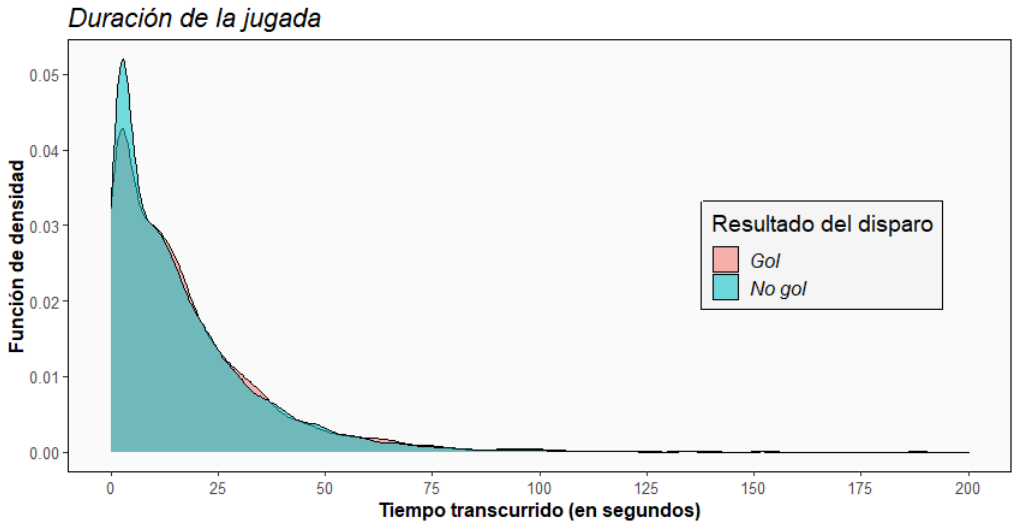

Figura 22: Función de densidad estimada para la duración de la jugada según el resultado del disparo.

Como se muestra en la Figura 22, la mayor parte de los tiros se realizan en jugadas que duran menos de 25 segundos, concretamente el 77.46 % del total. Las densidades en función del éxito o no del tiro son bastante similares, encontrándose una mayor diferencia en las jugadas que duran entre 0 y 10 segundos, aproximadamente. En ese intervalo se concentra el 39.06 % de los tiros no exitosos frente al 37.63 % de los que finalizan en gol. Tanto las medias como las medianas de ambas situaciones son muy cercanas. La mediana de los tiros fallidos es de 11.61 metros por los 11.98 metros de los tiros que acaban en gol. En el caso de la media los valores son 17.04 metros y 16.62 metros, respectivamente.

- **distanciaJugada**:
  Esta variable mide los metros que ha recorrido el balón durante toda la jugada. La medición se realiza a través de las coordenadas asociadas a los eventos de la jugada previos al tiro. Aquellos lanzamientos que son fruto de un balón suelto o rebote tienen un valor de *distanciaJugada* asociado de 0.0. Para el resto de casos, se recurre a los patrones de detección de inicio de la jugada, que como ya se mencionó anteriormente, se encuentran en el Anexo C.2.

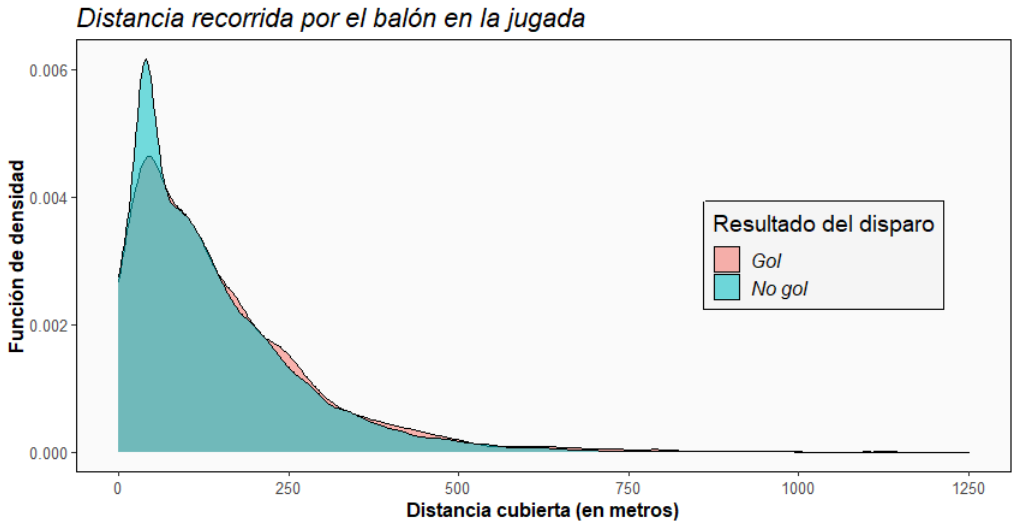

Figura 23: Función de densidad estimada para la distancia recorrida por el balón antes del golpeo en función del resultado del mismo.

En la Figura 23 podemos observar como las funciones de densidad para esta variable según el resultado del disparo son bastante similares. La mayor diferencia se encuentra en el intervalo [0,100], donde se producen el 42.67 % de los disparos sin éxito y el 38.61 % de los goles. Para valores fuera de ese intervalo, tenemos un alto grado de semejanza entre ambas funciones, existiendo pequeños intervalos en los que existen una mayor densidad de goles que de tiros fallidos, pero sin llegar al nivel de relevancia de lo observado en el intervalo [0,100].

- **velocidadJugada**:
  Esta variable se calcula como el cociente de *distanciaJugada* y *tiempoJugada* y, por tanto, se mide en *m/s*. Aquellos tiros que tienen un valor alto para este cociente significan que fueron consecuencia de un ataque más "eléctrico", jugadas de contra-ataque, que aquellos en los que se tienen valores bajos, como pueden ser las jugadas realizadas contra defensas ya posicionadas en el instante que se inicia la jugada.

  Para los tiros con $distanciaJugada = 0$ o $tiempoJugada = 0$ la variable *velocidadJugada* presenta un valor perdido (NA).

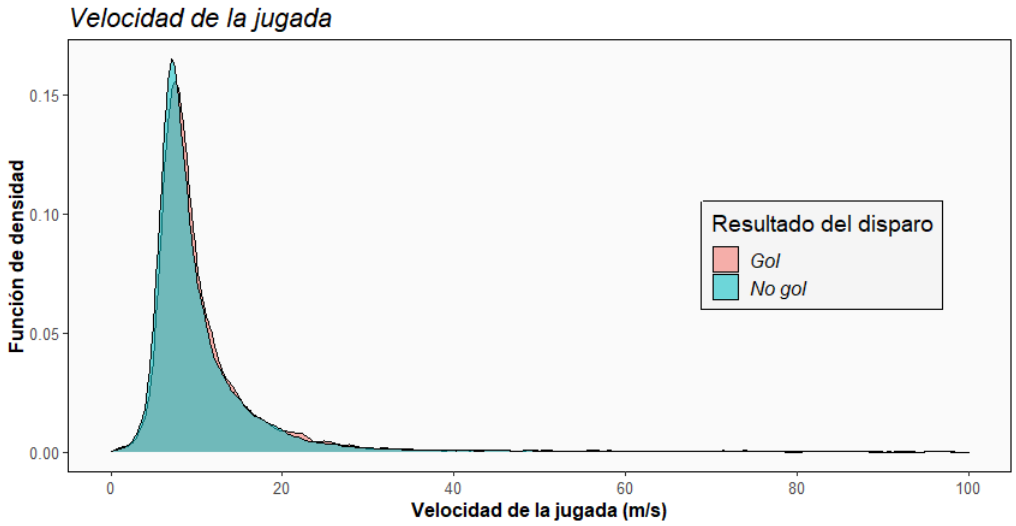

Figura 24: Función de densidad estimada para la velocidad de la jugada según el resultado del disparo.

Como se muestra en la Figura 24, la funciones de densidad son prácticamente idénticas. En el caso de los tiros fallidos, la mediana se encuentra situada en 8.36 m/s mientras que para los tiros que finalizan en gol se encuentra en 8.68 m/s. Si comparamos las medias de ambas situaciones encontramos valores algo más alejados pero que siguen siendo muy cercanos, 11.32 m/s en el caso de los fallidos y 12.69 m/s en el caso de los exitosos.

A pesar de que no parece tener una relación directa en el éxito de un tiro, se ha decidido mantener incluir esta variable ya que se considera que puede ser de utilidad su interacción con las demás.

6. Búsqueda del modelo XGBoost y resultados

En este capítulo se presenta la búsqueda de un modelo XGBoost para el problema, así como sus resultados. Los fundamentos teóricos de este algoritmo fueron presentados en el Capítulo 2.4. Se comienza definiendo el procedimiento seguido para encontrar el mejor modelo de goles esperados posible. Mediante una búsqueda aleatoria, dividida en dos fases, se determinan los hiper-parámetros más adecuados para el problema. Tras esto, se realiza un breve análisis exploratorio de los resultados obtenidos en cada fase de búsqueda, y una valoración del modelo final generado.

6.1. Creación del modelo

En esta sección se describe el proceso de búsqueda del mejor modelo posible de goles esperados. Para ello, se ha utilizado la librería XGBoost [23], ver Capítulo 2.4, en su versión para el software estadístico R.

Con el fin de poder utilizar esta librería, y como paso previo a la búsqueda del modelo, se ha tenido que realizar una adecuación del formato de los datos a los requerimientos técnicos de la librería. Las modificaciones realizadas sobre el conjunto de datos son:

- Conversión de todas las variables categóricas a numéricas, por medio de la librería *purrr* [24] y su función *map_df()*.
- Conversión de los datos a una matriz de tipo *DMatrix*.

Tras realizar ambas modificaciones, estamos en disposición de comenzar con la modelización. Dada la nula experiencia previa con problemas de estas características y con el uso de ensembles boosting, se ha optado por realizar una búsqueda aleatoria de los hiperparámetros del modelo.

6.1.1. ¿En qué consiste la búsqueda aleatoria de hiperparámetros?

Los hiperparámetros de un modelo son aquellos parámetros ajustables que permiten controlar su proceso de entrenamiento. La configuración de estos parámetros determinan en gran medida el rendimiento del modelo, por lo que su elección es de gran importancia. Existen dos formas de actuación a la hora de establecer estos parámetros.

La primera de ellas consiste en una *búsqueda manual* basada en el conocimiento y experiencia. Al ser un proceso manual, implica un alto coste tanto de tiempo como de recursos por lo que no es una opción muy recomendable si es la primera vez que te enfrentas a un problema de determinadas características.

La segunda opción es una *búsqueda aleatoria*, en la que no es preciso contar con un alto nivel de conocimiento ni experiencia previa. El primer paso para llevar a cabo un proceso como este es definir para cada hiperparámetro un rango de valores en los que se crea oportuno que pueda oscilar. Los rangos de valores establecidos para cada hiperparámetro definen lo que se conoce como el *espacio de búsqueda*. Tras definir este espacio, se realizan tantas combinaciones de hiperparámetros como se considere oportuno, utilizándose cada una de ellas para generar un nuevo modelo.

Una vez generados todos los modelos, se establece una comparación entre ellos mediante el error asociado a cada uno de ellos. La configuración de hiperparámetros que mejor resultado dé puede ser considerada como la óptima o bien puede ser utilizada para realizar una búsqueda manual a partir de ella.

6.1.2. Definición del espacio de búsqueda

Como se mencionó en la sección anterior, el espacio de búsqueda es el conjunto de todos los rangos de valores que pueden tomar los hiperparámetros considerados.

El primer paso para definir este espacio es determinar que hiperparámetros queremos utilizar. En nuestro caso, son los siguientes:

- **max_depth**: o profundidad máxima es el número de nodos máximo que puede haber entre el nodo raíz y cualquier otro nodo del árbol. Los árboles más profundos son más precisos, pero también tienen mayor riesgo de sobreajuste.
- **min_child_weight**: es el número mínimo de instancias necesarios para generar un nuevo nodo en el árbol. En este caso, a menor valor, mayor riesgo de sobreajuste.
- **subsample**: porcentaje de observaciones del total de las disponibles que se han de emplear en cada etapa de generación del modelo. De esta forma, se consigue que los sucesivos modelos no sean entrenados siempre con los mismo datos, sino que haya un poco de variedad a pesar de formar todos parte de un mismo conjunto.
- **colsample_bytree**: indica el número de variables que se van a utilizar en la generación de cada árbol. Al igual que el *subsample*, permite inducir cierta variabilidad.
- **eta**: es el parámetro de aprendizaje y determina el porcentaje de cambio con el que se actualizan los pesos en cada iteración. Valores bajos para este parámetro suelen proporcionar modelos más robustos al sobreajuste, pero también precisa de un mayor número de iteraciones.

Los rangos de valores escogidos para cada una de estos hiperparámetros pueden observarse en el Cuadro 16.

Hiperparámetro	Rango
max_depth	[5 , 12]
min_child_weight	[1 , 20]
subsample	[0.7 , 1.0]
colsample_bytree	[0.75 , 1.0]
eta	[0.01 , 0.3]

Cuadro 16: Definición del espacio de búsqueda.

6.1.3. Proceso de búsqueda

Además de definir el espacio de búsqueda es necesario fijar otros parámetros o argumentos que determinan el entrenamiento de los modelos:

- **booster**: es el tipo de clasificado base que se utilizará. En nuestro caso *gbtree*, que hace referencia a los árboles de decisión.
- **objective**: es la función objetivo, la que determina el tipo de aprendizaje que se va a llevar a cabo. En nuestro caso es *binary:logistic*
- **nrounds**: indica el número de iteraciones a realizar. En este trabajo se ha establecido a un valor de 300.
- **early_stopping_rounds**: número de iteraciones máximas a realizar sin conseguir mejora en la precisión del modelo. Lo establecemos en 35.
- **scale_pos_weight**: relación de instancias negativas (no-gol) por cada instancia positiva (gol). Permite dar un mayor peso a las instancias positivas en los datasets no balanceados, consiguiendo de esta forma un mejor ajuste. En nuestro caso, se ha fijado a 6.30.

Para medir la calidad del modelo se utilizará la métrica **AUC**, *Area Under the Curve*, que se calcula a partir de la *curva ROC* del modelo. La curva ROC es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales. Se basa en dos conceptos principales:

- *Tasa de Verdaderos Positivos (TPR)*: cociente del número de instancias clasificadas correctamente como positivas entre todas las que realmente lo son.
- *Tasa de Falsos Positivos (FPR)*: cociente del número de instancias clasificadas incorrectamente como positivas entre todas las que realmente son negativas.

La curva ROC enfrenta ambos valores, TPR en el eje X y FPR en el eje Y, para diferentes umbrales de clasificación, generando de esta forma una curva. El área encerrado bajo este área es a lo que denominamos AUC, y toma valores en el rango [0,1]. Los valores más próximos al 0 se dan en aquellos modelos cuyas predicciones son mayoritariamente erróneas, mientras que los valores cercanos a 1 se corresponden con modelos con un porcentaje de acierto muy elevado.

Una vez definido el espacio de búsqueda y el resto de parámetros necesarios, es momento de llevar a cabo la búsqueda del mejor modelo. El proceso está conformado por dos etapas:

1. Generación de 5.000 modelos mediante búsqueda aleatoria. Para agilizar este proceso, se realizará usando la técnica *Hold-Out* (2/3 - 1/3), a pesar de no ser la más recomendable en la optimización de árboles de decisión.

2. Evaluación de los 1.000 mejores modelos obtenidos, usando esta vez validación cruzada de 5 particiones.

Se ha de tener en cuenta que, para el proceso de creación del modelo, se van a utilizar, únicamente, los datos de los partidos disputados en las cinco grandes ligas europeas, mientras que los que hacen referencia a la Copa Mundial de la FIFA, celebrada en Rusia, son utilizados para evaluar la adecuación real del modelo implementado. Por tanto, la distribución de instancias para las distintas etapas, es la mostrada en el Cuadro 17.

Fase	**Nº instancias**	**% del total**
Entrenamiento	20401	63.86
Validación	10510	32.9
Test	1036	3.24
Total	31947	100

Cuadro 17: Distribución de las instancias para cada fase de creación del modelo.

Los porcentajes para entrenamiento y validación no son exactamente 2/3 - 1/3 como se dijo previamente. Esto es porque en el Cuadro 17 se tiene en cuenta el número total de instancias, mientras que la relación 2/3 - 1/3 hace referencia únicamente a las instancias que pueden ser empleadas para entrenamiento y validación, sin incluir las de test.

6.2. Resultados

En esta sección se va a realizar un breve análisis descriptivo de los resultados obtenidos para los modelos generados en la primera y segunda fase del proceso de búsqueda, prestando especial atención en el mejor modelo encontrado.

6.2.1. Evaluación primera fase de búsqueda

En esta fase se generaron 10.000 modelos mediante un proceso aleatorio de elección de hiperparámetros, en los rangos mostrados en el Cuadro 16. Se usó la técnica de Hold-Out para llevar a cabo la validación.

El proceso que se llevó a cabo con un procesador *i7* duró aproximadamente 10 horas. Los resultados obtenidos no arrojan grandes diferencias, como se puede observar en la Figura 25, dónde se muestra un extracto de las configuraciones consideradas, ordenadas de mejor a peor rendimiento.

	auc	booster	objective	max_depth	eta	subsample	colsample_bytree	min_child_weight
1	0.780100	gbtree	binary:logistic	5	0.10014691	0.7612183	0.8010451	2
2	0.780074	gbtree	binary:logistic	5	0.08252336	0.7157881	0.7683501	15
3	0.779941	gbtree	binary:logistic	5	0.07687709	0.9233183	0.8753567	20
4	0.779861	gbtree	binary:logistic	5	0.01771729	0.8515597	0.7850111	14
5	0.779749	gbtree	binary:logistic	5	0.02071593	0.8217831	0.8418789	20
6	0.779709	gbtree	binary:logistic	5	0.03050588	0.8063909	0.8934636	4
7	0.779679	gbtree	binary:logistic	5	0.06593117	0.7420265	0.8396573	5
8	0.779642	gbtree	binary:logistic	5	0.15562839	0.9514652	0.7625405	20
9	0.779634	gbtree	binary:logistic	5	0.02876594	0.8104867	0.8447888	8
10	0.779627	gbtree	binary:logistic	5	0.02245694	0.7098473	0.8376919	16
...	...	...	...	...	...	...	...	...
4997	0.735092	gbtree	binary:logistic	12	0.28500734	0.9948201	0.8867283	1
4998	0.734569	gbtree	binary:logistic	12	0.28650592	0.7844183	0.8931705	2
4999	0.734497	gbtree	binary:logistic	12	0.28598449	0.7434957	0.8879730	4
5000	0.733833	gbtree	binary:logistic	12	0.28711281	0.9863642	0.9340448	1

Figura 25: Extracto de los resultados obtenidos para los 10.000 modelos de búsqueda aleatoria.

Para el mejor modelo se tiene un **AUC** asociado de 0.7801 mientras que para el peor este valor es de 0.733833. Esta diferencia, de 0.037388 puntos, es muy pequeña teniendo en cuenta que los modelos han sido generados de forma aleatoria en unos rangos bastante amplios. El valor de la mediana para los AUC de los modelos se encuentra en el valor 0.767356, lo que significa que entre los mejores dos mil quinientos mejores modelos la diferencia, entre cualquiera de ellos, se traduce a menos de 1.275 puntos porcentuales. Esta diferencia entre los dos mil quinientos peores se multiplica hasta los 3.35 puntos.

Si nos fijamos en los hiper-parámetros destaca la relevancia de **max_depth**. De los 100 mejores modelos obtenidos, el 96 % tienen una profundidad máxima igual 5, habiendo únicamente cuatro con un profundidad de 6. En el Cuadro 18 se muestra la distribución de esta variable en función de los cuartiles de los modelos con mejor rendimiento.

max_depth/ranking	**1-25 %**	**25-50 %**	**50-75 %**	**75-100 %**
5	613	7	0	0
6	431	196	4	0
7	144	363	112	1
8	49	262	230	56
9	8	171	272	184
10	3	126	232	259
11	1	82	218	354
12	1	43	182	396

Cuadro 18: Distribución de las profundidades de los modelos en función de su ranking.

Existe una clara tendencia entre niveles de profundidad más bajos y el ranking del modelo. Entre los 1250 mejores modelos, 613, prácticamente la mitad, tienen una profundidad máxima de 5. Si nos fijamos en los valores más elevados, vemos que la mayor parte se dan en los modelos que peor clasifican. Por ejemplo, de los 1250 peores modelos, 396, el 31.68 %, tienen una profundidad máxima igual a 12.

Respecto al hiper-parámetro **min_child_weight**, la mediana de los 1250 mejores modelos se sitúa en 12, lo que indica un mejor rendimiento para valores en el intervalo [12,20] que en el [1,12]. Si nos fijamos en los 1250 con peores resultados, la mediana baja hasta 7, lo que evidencia que valores bajos para esta variable dan peor resultado.

Para el hiper-parámetro de aprendizaje, **eta**, apreciamos que los valores más bajos proporcionan mejores resultados. En los mil doscientos cincuenta mejores modelos, teniendo en cuenta que el intervalo para este parámetro (ver Cuadro 16) se definió en [0.01, 0.3], la mediana se sitúa entorno a 0.113, bastante más cercano al límite inferior del intervalo que al superior. Por el contrario, la mediana de este valor para los 1250 peores modelos, se sitúa alrededor de 0.2229.

Por último, para los hiper-parámetros **subsample** y **colsample_bytree**, no existen grandes diferencias detectadas. En el primer caso, la distribución es muy similar lo que lleva a pensar que no es relevante, al menos, en presencia del resto de hiper-parámetros. En el caso de *colsample_bytree* los cinco mil mejores modelos presentan su mediana entorno a 0.866 lo que, dado su intervalo (ver Cuadro 16), significa que hay una cierta preferencia por valores bajos, aunque no es demasiado clara, ya que la mediana de los cinco mil peores se sitúa alrededor 0.882.

Teniendo en cuenta todos estos aspectos, y como parte del proceso de búsqueda descrito en el Capítulo 6.1.3, se va a llevar a cabo un procedimiento de validación cruzada, con cinco particiones, sobre las 1000 mejores configuraciones, con el objetivo de determinar el modelo más apropiado.

6.2.2. Evaluación segunda fase de búsqueda

En esta fase se utilizan las 1000 mejores configuraciones obtenidas para entrenar nuevos modelos y evaluarlos mediante la técnica de validación cruzada de cinco particiones. Los resultados obtenidos se muestran en la Figura 26.

	auc	booster	objective	max_depth	eta	subsample	colsample_bytree	min_child_weight
1	0.7725842	gbtree	binary:logistic	5	0.05681972	0.8509507	0.7580628	12
2	0.7724344	gbtree	binary:logistic	5	0.04142479	0.9601867	0.8763695	19
3	0.7722142	gbtree	binary:logistic	5	0.02245694	0.7098473	0.8376919	16
4	0.7721210	gbtree	binary:logistic	6	0.03023359	0.9608868	0.8616448	18
5	0.7721138	gbtree	binary:logistic	5	0.03273589	0.9423410	0.7686803	20
6	0.7719372	gbtree	binary:logistic	5	0.04300865	0.7573266	0.8274396	7
7	0.7719090	gbtree	binary:logistic	5	0.05777612	0.7952564	0.8397107	15
8	0.7718118	gbtree	binary:logistic	5	0.02653896	0.8268575	0.9222693	16
9	0.7718104	gbtree	binary:logistic	5	0.05340720	0.8723558	0.9275230	16
10	0.7717938	gbtree	binary:logistic	6	0.01560691	0.8630827	0.8652220	12
...	...	...	...	...	...	...	...	...
997	0.7615776	gbtree	binary:logistic	6	0.27215442	0.7917928	0.9029441	9
998	0.7615286	gbtree	binary:logistic	6	0.18500694	0.7452080	0.8866726	6
999	0.7610572	gbtree	binary:logistic	6	0.20418488	0.8352573	0.8579500	1
1000	0.7599706	gbtree	binary:logistic	8	0.08997364	0.8762503	0.7674509	15

Figura 26: Extracto de los resultados obtenidos para los 1.000 mejores modelos de búsqueda aleatoria.

El mejor modelo tiene un área bajo la curva de 0.7725842, mientras que en el peor es de 0.75997, aproximadamente. Esta diferencia es mayor que la conseguida para estos mil modelos obtenidos mediante Hold-Out. Esto se debe a que ahora, al usar validación cruzada, se cuenta con un mayor número de instancias para realizar el entrenamiento, lo que aumenta la capacidad de aprendizaje ya que el AUC no se calcula haciendo uso de un pequeño conjunto de instancias.

Respecto a las hiper-parámetros continuos, **eta**, **subsample** y **colsample_bytree**, obtenemos las mismas conclusiones que en la sección anterior. Los mejores resultados se obtienen con un ratio de aprendizaje bajo, cercano al límite inferior del intervalo establecido (ver Cuadro 16). Por su parte, subsample y colsample_bytree, siguen distribuciones muy similares en los diferentes tramos del ranking de los modelos.

Para el hiper-parámetro **max_depth** se vuelve a observar una clara preferencia por los valores más bajos, como puede verse en el Cuadro 19.

max_depth/ranking	**1-25 %**	**25-50 %**	**50-75 %**	**75-100 %**
5	204	157	133	87
6	44	81	86	114
7	1	9	26	36
8	1	3	5	13

Cuadro 19: Distribución de las profundidades de los 1000 mejores modelos en función de su ranking.

Hay que tener en cuenta que la frecuencia de estos valores no es uniforme, como en el caso de la primera búsqueda, sino que ahora más de la mitad corresponden con una profundidad máxima de 5, mientras que solo hay dos y tres modelos con una profundidad máxima de 9 y 10 nodos, respectivamente. Aún así, sigue percibiéndose una tendencia clara de preferencia por árboles con profundidades bajas.

Por último, para el hiper-parámetro **min_child_weight** observamos, también, un comportamiento similar al descrito en la primera fase. En el percentil 25 de los mejores modelos, la concentración de valores próximos al centro del intervalo de búsqueda (ver Cuadro 16) es superior a cualquier otro tramo del ranking. Lo que denota una preferencia por valores medios, más que extremos. En el resto de tramos del ranking, la concentración de valores extremos va en aumento.

6.2.3. Evaluación modelo final para los partidos del Mundial

Una vez realizado el proceso de búsqueda y tras valorar los resultados, se ha decidido utilizar la mejor configuración obtenida en la segunda fase para construir el modelo final. Los hiper-parámetros escogidos, como puede verse en la Figura 26, son:

- **max_depth**: 5
- **eta**: 0.05681972
- **subsample**: 0.8509507
- **colsample_bytree**: 0.7580628
- **min_child_weight**: 12

Para el modelo obtenido con la configuración descrita, y las consideraciones hechas en el Capítulo 6.1.3, la importancia de cada variable (ver Capítulo 5) se puede observar en la Figura 27.

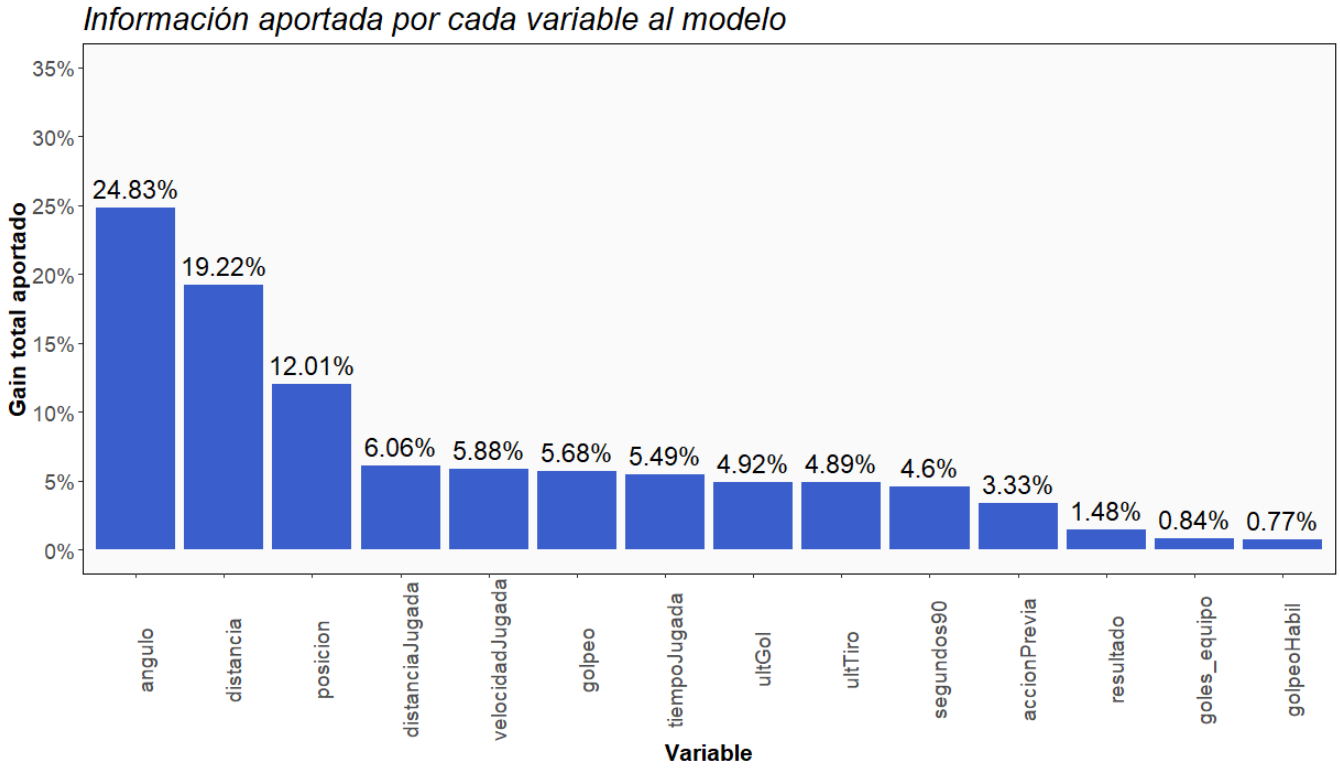

Figura 27: Información aportada por cada variable al modelo.

Las tres variables que más información aportan al modelo son *ángulo*, *distancia* y *posición*, tal y cómo se podía suponer con las consideraciones vistas en el Capítulo 4.1.1. Entre las tres, aportan el 56.06 %.

Existen otras cuatro variables con un nivel de aportación superior al 5 %, y son: *distanciaJugada*, *velocidadJugada*, *golpeo* y *tiempoJugada*. Destaca la baja aportación de *segundos90* y la casi nula de *resultado* y *goles_equipo* (ver Figura 27), lo que indica que el factor psicológico no es tan determinante en nuestro modelo, en presencia del resto de variables. Este hecho confronta con la idea inicial de que la presión a la hora de realizar el lanzamiento tuviese un impacto en su probabilidad de éxito.

Pero, ¿cómo de preciso es el clasificador? Esta es una pregunta difícil de responder, ya que el modelo calcula para cada instancia (tiro) una probabilidad de éxito, pero no porque ésta tenga un valor más elevado quiere decir que realmente pertenezca a la clase positiva (gol). Es factible, que un tiro con una probabilidad de acabar en gol de 0.73 realmente no haya finalizado con éxito, mientras que otro con una probabilidad de 0.27 sí lo haya hecho.

Como se mencionó en el Capítulo 4, es esperable que un modelo de goles esperados produzca mejores resultados en valoraciones amplias, en las que se contemple muchos tiros o partidos, que en otras más concretas, en las que solo se valoren unos pocos encuentros. Por ello, una forma de medir la precisión del modelo es comparar el número total de goles esperados con los realmente acontecidos. Cuanto más próximos sean ambos valores, mas preciso es el modelo.

El cálculo de los goles esperados bajo el modelo se resume en la suma de las probabilidades asignadas a las instancias de test, que son los tiros considerados para el Mundial de 2018. En el Cuadro 20 se muestra el xG obtenido, el número de tiros en el conjunto y el número de goles considerados.

xG Modelo	**Goles considerados**	**Tiros considerados**
363.049	128	1036

Cuadro 20: Comparación número de goles esperados y total considerados.

El valor de los goles esperados es casi el triple de los realmente acontecidos, lo que hace indicar que nuestro modelo no está llevando a cabo una buena clasificación. Aún así, hay que tener en cuenta que en se trabaja en un escenario de alta incertidumbre, por lo que es posible que si se contase con un mayor número de instancias, la aproximación realizada se estabilizase en valores más cercanos al real.

Para tratar de arrojar más luz sobre el comportamiento del modelo, utilizamos las métricas binarias. Para su utilización, cada instancia es catalogada como "gol" o "no-gol" en función de la comparación de su probabilidad asignada con un umbral establecido, denominado *threshold*. Los resultados obtenidos son los que se muestran en el Cuadro 21.

Threshold	**TP**	**TN**	**FP**	**FN**	**Sensibilidad**	**Especificidad**	**balancedAccuracy**
0.1	124	146	762	4	0.969	0.161	0.261
0.2	115	345	563	13	0.898	0.38	0.444
0.3	106	500	408	22	0.828	0.551	0.585
0.4	100	604	304	28	0.781	0.665	0.68
0.5	85	702	206	43	0.664	0.773	0.76
0.6	64	779	129	64	0.5	0.858	0.814
0.7	46	858	50	82	0.359	0.945	0.873
0.8	20	892	16	108	0.156	0.982	0.88
0.9	8	903	5	120	0.062	0.994	0.879

Cuadro 21: Resumen métricas binarias.

La columna *TP (True Positive)* (ver Cuadro 21) hace referencia a las instancias bien clasificadas. Vemos cómo según aumentamos el threshold este valor va disminuyendo, provocando el concordante aumento del número de falsos negativos representado en la columna *FN (False Negative)*. Mediante la columna *Sensibilidad (o TPR)* tenemos la probabilidad de clasificar correctamente una instancia positiva (ver Capítulo 6.1.3), en nuestro caso un gol, para cada nivel del umbral. Los mejores resultados se consiguen con umbrales bajos, hasta tal punto que con un *threshold* de 0.6, se clasifican bien la mitad de ellas.

La *Especificidad* es el valor complementario del *FPR (False Positive Rate)* visto en el Capítulo 6.1.3. Mide la probabilidad de que una instancia negativa, en nuestro caso "no-gol", sea clasificada como tal. Su cálculo se reduce a la división del número de instancias clasificadas correctamente como negativa, columna *TN (True Negative)*, entre el número total de instancias clasificadas como negativas, columnas *TN* y *FP (False Positive)*.

La elección del mejor umbral suele hacerse en función de los valores de sensibilidad y especificidad. Normalmente, se busca que exista un cierto equilibrio entre ambos, pero en otras ocasiones, como en la detección de enfermedades graves, se prefiere una mayor especificidad a cambio de perder sensibilidad. En nuestro caso, podemos considerar como óptimos los umbrales 0.4 y 0.5 que aportan cierto equilibrio entre ambas métricas (ver Cuadro 21).

La precisión del modelo para esos umbrales es de 0.68 y 0.76, respectivamente, como podemos ver en la columna *balancedAccuracy* (ver Cuadro 21), que mide el ratio de instancias bien clasificadas. No son valores muy elevados, lo que indica que el modelo no clasifica todo lo bien que desearíamos. Existen umbrales para los que la precisión es bastante alta, como ocurre con los thresholds de 0.7, 0.8 y 0.9, pero esto es a costa de reducir notablemente la sensibilidad, algo no deseado.

Haciendo uso de los valores vistos para la sensibilidad y especificidad, se representa la curva ROC en la Figura 28.

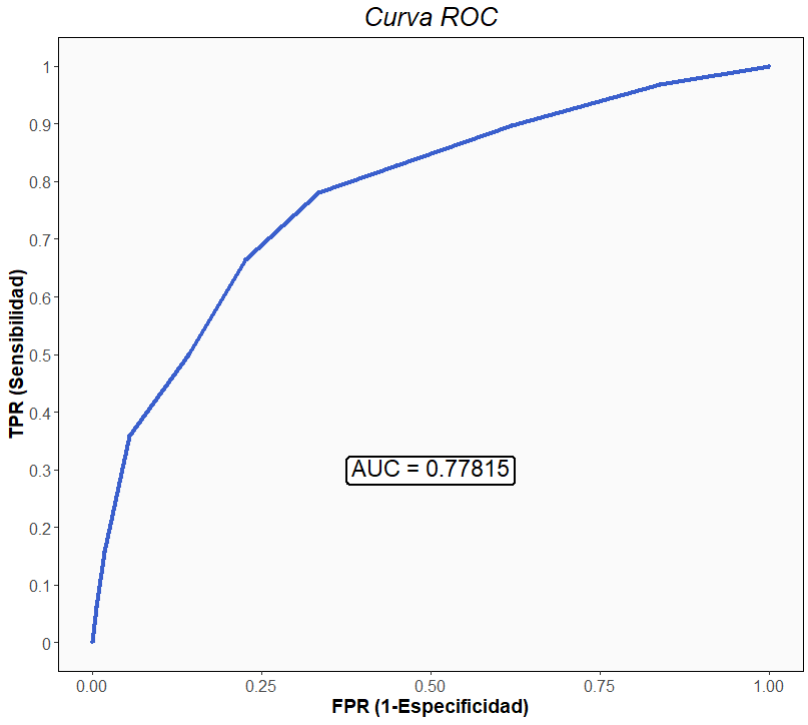

Figura 28: Curva ROC del modelo final con los datos del Mundial de Rusia de 2018.

Como podemos ver en la Figura 28, el modelo construido tiene un AUC de 0.77815, lo que indica que no es un mal clasificador. Sin embargo, como vimos en el Cuadro 20, las probabilidades asignadas a cada tiro no se ajustan bien al objetivo del problema, por lo que no es un modelo del todo adecuado para el objetivo que se perseguía.

Esto puede deberse al uso de la métrica AUC para evaluar el rendimiento de los modelos. Esta métrica se centra más en obtener un modelo preciso que en proporcionar unas probabilidades fieles al objetivo planteado. Por ejemplo, sea la variable respuesta para cuatro instancias (1,1,0,1) y las probabilidades de pertenencia asignadas por dos modelos distintos: (0.51,0.51,0.31,0.51) y (0.9,0.86,0.2,0.7). En este caso, con un threshold de 0.5, ambos tienen la misma precisión a pesar de que las probabilidades son muy diferentes.

7. Conclusiones

En este Trabajo Fin de Grado se ha construido un modelo de goles esperados para la Copa Mundial de la FIFA del año 2018, tomando como punto de partida el conjunto de datos proporcionado por *WyScout* [3].

Dado que el conjunto de datos contenía la información en crudo de todos los partidos disputados en las cinco grandes ligas europeas de la temporada 2017-2018 y en el Mundial de 2018, fueron varios los objetivos a cubrir antes de crear el modelo.

El primero de los objetivos que se afrontó fue la re-estructuración de la información contenida en el conjunto de datos original. Los ficheros presentaban una organización compleja, por lo que comprender su estructura y adaptarla a las necesidades de nuestro problema constituyó un paso fundamental para la evolución del proyecto. Se utilizaron herramientas no vistas durante la formación académica, como la librerías *jsonlite* y *dplyr* del software R, que permitieron explorar a fondo la información contenida en los datos y adaptarla a un esquema de trabajo más cómodo, como se describió en el Capítulo 3.

El segundo de los objetivos fijados consistió en establecer una comparación entre las cinco grandes ligas europeas, que justificase su uso como espacio muestral representativo para la Copa Mundial de la FIFA. Mediante las comparación de las nacionalidades de sus jugadores y la evolución de los resultados de los partidos, en función de la condición de local o visitante del equipo que comienza marcando, se pudieron encontrar bastantes símiles y diferencias entre ellas, que justifican su utilización para la creación del modelo. Esto se realizó en el Capítulo 4.

El tercer objetivo, y el más relevante de los vistos hasta ahora para la creación del modelo, fue la generación de las variables a considerar. Supuso el proceso más largo y tedioso del proyecto ya que, además de una fase de investigación sobre otros modelos, hubo que tratar con los datos "en crudo" para la obtención de información relevante. De esta forma, se definieron un total de 15 variables explicativas, por medio de procesos de distinta dificultad exploratoria.

Como se puede juzgar por los resultados mostrados en el Capítulo 6.2, la información contenida en las variables creadas es bastante útil. Quiero destacar en este punto, la relevancia vista para las variables *distanciaJugada*, *tiempoJugada* y *velocidadJugada*, en el modelo final (ver Figura 27). Estas tres variables fueron las más difíciles de definir, y aportan una mayor nivel de información que otras como *golpeo* que puede considerarse en un primer momento como más determinante.

Para la creación del modelo de goles esperados se optó por un algoritmo de *Gradient Boosting*, concretamente su implementación *XGBoost*. La elección se tomó con el fin de cubrir un área de conocimiento que no había sido visto con detalle durante la formación académica y que consideraba de gran interés. Desde ese punto de vista, el trabajo ha supuesto un gran aprendizaje tanto teórico como práctico.

En cuanto a la consecución del objetivo principal del proyecto haciendo uso de dicho algoritmo, no se han obtenido los resultados deseados. Si bien es cierto que se ha podido desarrollar un clasificador binario con un buen nivel de precisión, las probabilidades asignadas a cada tiro están lejos de poder ser consideradas como aceptables. En total, según el modelo, durante el Mundial de Rusia de 2018 se generaron ocasiones como para conseguir 363 goles en sus 64 partidos, a una media de 5.67 por encuentro.

La métrica para escoger el mejor modelo fue *AUC*, que mide la precisión del clasificador. Es posible que esta elección es la que haya lastrado los resultados obtenidos para las probabilidades, ya que hace más hincapié en conseguir un clasificador con la mayor precisión posible que en dar unas probabilidades fieles a nuestro objetivo.

Las principales líneas futuras de trabajo se establecen entorno a la mejora de los resultados:

- **Realizar el proceso de búsqueda con otra función de evaluación**, que se centre más en obtener unas probabilidades óptimas y no tanto en la precisión del modelo. Por ejemplo, la función *logloss* podría ser un buen comienzo.
- **Incluir una nueva fase en el proceso de búsqueda**, de carácter más experimental, en la que tras analizar el comportamiento de los modelos generados, se lleve a cabo un ajuste manual de los hiper-parámetros que permita obtener un mejor modelo.
- **Generar el modelo bajo otros algoritmos**, como pueden ser otras implementaciones del Gradient Boosting, como *LightGBM*, o el uso de redes neuronales.

Otras mejoras futuras pueden ser la creación de nuevas variables, como el número de ataques del equipo que realiza el tiro en los cinco minutos previos, o el desarrollo de una aplicación Shiny que permita explorar los resultados obtenidos con el modelo de forma visual, seleccionando el partido del que se desea ver la información.

Anexos

A. Anexo I: Información adicional ficheros WyScout

En este Anexo se muestran los significados de las codificaciones numéricas de las columnas *eventId*, *subEventId* y *tags_id* de los ficheros events_X.json proporcionados por WyScout.

A.1. Relación de eventos y subeventos en ficheros events_X.json

Todos y cada uno de los eventos contenidos en los ficheros *events_X.json* están categorizados en función de su naturaleza. En total, existen 10 tipos de eventos distintos, que dan lugar hasta 60 subeventos diferentes. Además, cada uno de estos eventos y subeventos tienen asociado un código numérico que permite manipular el conjunto de datos de forma más sencilla. Las categorías de eventos existentes, y sus códigos numéricos, se muestran en la Figura 29.

event	subevent	event_label	subevent_label
1	10	Duel	Air duel
1	11	Duel	Ground attacking duel
1	12	Duel	Ground defending duel
1	13	Duel	Ground loose ball duel
2	20	Foul	Foul
2	21	Foul	Hand foul
2	22	Foul	Late card foul
2	23	Foul	Out of game foul
2	24	Foul	Protest
2	25	Foul	Simulation
2	26	Foul	Time lost foul
2	27	Foul	Violent Foul
3	30	Free Kick	Corner
3	31	Free Kick	Free Kick
3	32	Free Kick	Free kick cross
3	33	Free Kick	Free kick shot
3	34	Free Kick	Goal kick
3	35	Free Kick	Penalty
3	36	Free Kick	Throw in
4	40	Goalkeeper leaving line	Goalkeeper leaving line
5	50	Interruption	Ball out of the field
5	51	Interruption	Whistle
6	60	Offside	Offside
7	70	Others on the ball	Acceleration
7	71	Others on the ball	Clearance
7	72	Others on the ball	Touch
8	80	Pass	Cross
8	81	Pass	Hand pass
8	82	Pass	Head pass
8	83	Pass	High pass
8	84	Pass	Launch
8	85	Pass	Simple pass
8	86	Pass	Smart pass
9	90	Save attempt	Reflexes
9	91	Save attempt	Save attempt
10	100	Shot	Shot

Figura 29: Relación de eventos y subeventos.

Los nombres otorgados son bastante representativos, pero si se quiere indagar más sobre alguno de ellos se puede consultar el glosario de términos de *WyScout* [16].

A.2. Interpretación columnas tags_id en ficheros events_X.json

Para cada evento de los ficheros events_X.json se cuenta con información sobre cómo se produjo y el impacto que tuvo en el desarrollo del partido. Para cualquier evento dado podemos encontrarnos con hasta seis valores asociados que aportan dicha información. Por ejemplo, en el caso de un pase estas etiquetas nos permiten conocer si el pase fue preciso o no, con que pie se realizó, si supuso una pérdida peligrosa de balón, etcétera.

A continuación, en las Figuras 30 y 31, se muestra el listado con todos los posibles valores asociados a un evento y su significado.

Tag	Label	Description
101	Goal	Goal
102	own_goal	Own goal
301	assist	Assist
302	keyPass	Key pass
1901	counter_attack	Counter attack
401	Left	Left foot
402	Right	Right foot
403	head/body	Head/body
1101	direct	Direct
1102	indirect	Indirect
2001	dangerous_ball_lost	Dangerous ball lost
2101	blocked	Blocked
801	high	High
802	low	Low
1401	interception	Interception
1501	clearance	Clearance
201	opportunity	Opportunity
1301	Feint	Feint
1302	missed ball	Missed ball
501	free_space_r	Free space right
502	free_space_l	Free space left
503	take_on_l	Take on left
504	take_on_r	Take on right
1601	sliding_tackle	Sliding tackle
601	anticipated	Anticipated
602	anticipation	Anticipation
1701	red_card	Red card
1702	yellow_card	Yellow card
1703	second_yellow_card	Second yellow card
1201	gb	Position: Goal low center
1202	gbr	Position: Goal low right

Figura 30: Etiquetas que pueden ser asociadas a un evento (I).

Tag	Label	Description
1203	gc	Position: Goal center
1204	gl	Position: Goal center left
1205	glb	Position: Goal low left
1206	gr	Position: Goal center right
1207	gt	Position: Goal high center
1208	gtl	Position: Goal high left
1209	gtr	Position: Goal high right
1210	obr	Position: Out low right
1211	ol	Position: Out center left
1212	olb	Position: Out low left
1213	or	Position: Out center right
1214	ot	Position: Out high center
1215	otl	Position: Out high left
1216	otr	Position: Out high right
1217	pbr	Position: Post low right
1218	pl	Position: Post center left
1219	plb	Position: Post low left
1220	pr	Position: Post center right
1221	pt	Position: Post high center
1222	ptl	Position: Post high left
1223	ptr	Position: Post high right
901	through	Through
1001	fairplay	Fairplay
701	lost	Lost
702	neutral	Neutral
703	won	Won
1801	accurate	Accurate
1802	not accurate	Not accurate

Figura 31: Etiquetas que pueden ser asociadas a un evento (II).

Tras el proceso descrito en el Capítulo 3.4 cada evento cuenta con uno de estos valores en sus columnas *tags_idX*. Para profundizar en el significado de estas etiquetas se puede consultar el glosario de *WyScout* [16].

B. Anexo II: Creación representaciones gráficas Capítulo 4

En este Anexo se explican los procesos de recopilación de información y tratamiento de datos realizados para poder generar cada una de las representaciones gráficas vistas en el Capítulo 4.

B.1. Mapas cartográficos y jugadores por categoría

Para la realización de las Figuras 6 y 7 se ha realizado un proceso manual de recolección de datos y posteriormente se han empleados los paquetes de R *rnaturalearth* [25] y *ggplot2* [26] para crear las representaciones gráficas.

La información se ha obtenido a través de las convocatorias oficiales de cada una de los combinados nacionales que disputaron el torneo, y que la *FIFA* publicó de manera conjunta [27]. En dicha publicación, para cada selección se detalla el dorsal, la posición, el nombre, la fecha de nacimiento, el nombre de la camiseta, el club, la altura y el peso de cada uno de sus jugadores. Por ejemplo, en la Figura 32, se muestra la convocatoria de la selección argentina.

Team	#	Pos.	FIFA Popular Name	Birth Date	Shirt Name	Club	Height	Weight
Argentina	1	GK	GUZMAN Nahuel	10.02.1986	GUZMÁN	Tigres UANL (MEX)	192	90
Argentina	2	DF	MERCADO Gabriel	18.03.1987	MERCADO	Sevilla FC (ESP)	181	81
Argentina	3	DF	TAGLIAFICO Nicolas	31.08.1992	TAGLIAFICO	AFC Ajax (NED)	169	65
Argentina	4	DF	ANSALDI Cristian	20.09.1986	ANSALDI	Torino FC (ITA)	181	73
Argentina	5	MF	BIGLIA Lucas	30.01.1986	BIGLIA	AC Milan (ITA)	175	73
Argentina	6	DF	FAZIO Federico	17.03.1987	FAZIO	AS Roma (ITA)	199	85
Argentina	7	MF	BANEGA Ever	29.06.1988	BANEGA	Sevilla FC (ESP)	175	73
Argentina	8	DF	ACUNA Marcos	28.10.1991	ACUÑA	Sporting CP (POR)	172	77
Argentina	9	FW	HIGUAIN Gonzalo	10.12.1987	HIGUAÍN	Juventus FC (ITA)	184	75
Argentina	10	FW	MESSI Lionel	24.06.1987	MESSI	FC Barcelona (ESP)	170	72
Argentina	11	MF	DI MARIA Angel	14.02.1988	DI MARÍA	Paris Saint-Germain FC (FRA)	178	75
Argentina	12	GK	ARMANI Franco	16.10.1986	ARMANI	CA River Plate (ARG)	189	85
Argentina	13	MF	MEZA Maximiliano	15.12.1992	MEZA	CA Independiente (ARG)	180	76
Argentina	14	DF	MASCHERANO Javier	08.06.1984	MASCHERANO	Hebei China Fortune FC (CHN)	174	73
Argentina	15	MF	LANZINI Manuel	15.02.1993	LANZINI	West Ham United FC (ENG)	167	66
Argentina	16	DF	ROJO Marcos	20.03.1990	ROJO	Manchester United FC (ENG)	189	82
Argentina	17	DF	OTAMENDI Nicolas	12.02.1988	OTAMENDI	Manchester City FC (ENG)	181	81
Argentina	18	DF	SALVIO Eduardo	13.07.1990	SALVIO	SL Benfica (POR)	167	69
Argentina	19	FW	AGUERO Sergio	02.06.1988	AGÜERO	Manchester City FC (ENG)	172	74
Argentina	20	MF	LO CELSO Giovani	09.04.1996	LO CELSO	Paris Saint-Germain FC (FRA)	177	75
Argentina	21	FW	DYBALA Paulo	15.11.1993	DYBALA	Juventus FC (ITA)	177	73
Argentina	22	MF	PAVON Cristian	21.01.1996	PAVÓN	CA Boca Juniors (ARG)	169	65
Argentina	23	GK	CABALLERO Wilfredo	28.09.1981	CABALLERO	Chelsea FC (ENG)	186	80

Figura 32: Jugadores convocados por la selección argentina.

Tras recorrer todas las convocatorias y teniendo en cuenta ciertos aspectos, como que el *Swansea City AFC* es un equipo galés que compite en el sistema de competición inglés, obtenemos para cada país el número de jugadores asociado a sus sistemas de competición (ver Cuadro 22).

País	**Nº de jugadores**
England	130
Spain	81
Germany	67
Italy	58
France	49
Russia	36
Saudi Arabia	30
Mexico, Turkey	22
Portugal	19
United States of America	18
Belgium	16
Netherlands, Japan	15
South Korea	13
Egypt	10
Argentina, Brazil , Iran	9
China, Denmark	8
Peru, Costa Rica, Scotland	7
Colombia, Tunisia	6
Greece	5
Switzerland, Poland	4
Panama, Ukraine, Sweeden, Serbia, Croatia, Australia	3
Guatemala, Chile, Israel, Austria, Uruguay, Morocco, Qatar, Bulgaria, United Arab Emirates	2
Slovakia, Honduras, Romania, South Africa, Canada, Guinea, Iceland, Norway, Nigeria, Finland	1

Cuadro 22: Ligas de proveniencia de los jugadores que participaron en el campeonato del mundo del año 2018.

A partir de esta tabla obtenemos las frecuencias correspondientes, que son los valores que se representan en los mapas de las Figuras 6 y 7. Haciendo uso de la función *ne_countries*, del paquete *rnaturalearth*, obtenemos los polígonos que representan los territorios de cada uno de los países y usando la librería *ggplot* se generan ambas representaciones. A continuación se muestra el código empleado.

```
# Poligonos de los paises
world<-ne_countries(type='map_units',returnclass ="sf")%>%
filter(name_sort!="Antarctica")%>%fortify
playersFromLeagues<-rep(NA,182)
world<-cbind(world,playersFromLeagues)

playersByLeague<-data.frame(region=c('England','Spain',
'Germany','Italy','France','Russia','Saudi Arabia',
'Mexico','Turkey','Portugal',
'United States of America','Belgium','Netherlands',
'Japan','South Korea','Egypt','Argentina','Brazil',
'Iran','China','Denmark','Peru','Costa Rica',
'Scotland','Colombia','Tunisia','Greece',
'Switzerland','Poland','Panama','Ukraine',
'Sweeden','Serbia','Croatia','Australia',
'Guatemala','Chile','Israel','Austria','Uruguay',
'Morocco','Qatar','Bulgaria','United Arab Emirates',
'Slovakia','Honduras','Romania','South Africa',
'Canada','Guinea','Iceland','Norway','Nigeria',
'Finland'),
value=c(130,81,67,58,49,36,30,22,22,19,18,16,
15,15,13,10,9,9,9,8,8,7,7,7,6,
6,5,4,4,3,3,3,3,3,3,2,2,
2,2,2,2,2,2,2,1,1,1,1,
1,1,1,1,1,1)/736*100,
stringsAsFactors = FALSE)

# Relacionar cada poligono con el valor correspondiente
for(geo in 1:nrow(world)){
        geounit<-world$geounit[geo]
        for(reg in 1:dim(playersByLeague)[1]){
                if(geounit==playersByLeague$region[reg]){
                        val<-playersByLeague$value[reg]
                        world$playersFromLeagues[geo]<-val
                }
        }
}

# Representacion cartografica Europa
```

```
europe <- world[world$region_un=="Europe"&
world$name!='Russia',]

cad<-"+proj=aea +lat_1=36.333333333333336
+lat_2=65.66666666666667 +lon_0=14"

ggplot(data=europe)+
geom_sf(aes(fill=playersFromLeagues)) +
scale_fill_continuous(breaks = c(17, 12, 7, 2),
labels = c("< 17%","< 12%", "< 7%", "< 2%"),
low="wheat1", high="firebrick2",
na.value="lightyellow",
name="Porcentaje\n") +
guides(fill = guide_colourbar(barwidth = 0.6,
barheight = 5, ticks = FALSE))+
theme(
panel.background =
element_rect(fill = "slategray1"),
panel.border = element_rect(fill = NA),
axis.text.y = element_blank(),
axis.ticks.y = element_blank(),
axis.text.x = element_blank(),
axis.ticks.x = element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
legend.title = element_text("Porcentaje"),
legend.background =
element_rect(fill="gray96"),
legend.position = c(0.87,0.65))+
coord_sf(crs=cad)
```

Para el mapa mundial se emplea el mismo código *ggplot* que para el mada de Europa pero con el data frame *world* en lugar de *europe* y sin la opción *coord_sf*.

Para la realización de la Figura 22 se ha empleado la misma fuente de datos y para cada equipo asociado a los sistemas de competición de las cinco grandes ligas se ha buscado la categoría en la que militaban durante esa campaña, y posteriormente con *ggplot* se ha generado el gráfico.

B.2. Gráficos evolución de resultados

Para la elaboración de las Figuras 9, 10, 11 y 12, se crean dos nuevos data frames a partir de los conjuntos: *matches.csv*, *events.csv* y *teams.csv*.

El nuevo data frame, denominado *goals.csv*, contiene la información de todos los eventos que se tradujeron en gol en cada una de las seis competiciones. Para generarlo se empleó el fichero *events.csv* y a través de las *tags* asociadas a cada evento se determinó si la acción finalizaba en gol o no. Las etiquetas que nos interesan en este caso son *101* (gol) y *102* (gol en propia puerta). Además, los eventos del tipo *Save attempt* fueron excluidos ya que a pesar de compartir estas etiquetas se trata de un tipo de evento asociado a un tiro o acción que finaliza en gol, por lo que es información repetida. Para cada evento de interés guardamos toda la información correspondiente contenida en *events.csv*. En la Figura 33 se muestran las 8 primeras entradas.

	X	id	matchId	teamId	playerId	eventName	eventId	subEventName	subEventId	eventSec	matchPeriod
1	1	180866021	2565548	695	225089	Free Kick	3	Penalty	35	2571.81857	2H
2	2	180054901	2565549	692	395636	Shot	10	Shot	100	1300.58282	1H
3	3	180055162	2565549	687	355599	Shot	10	Shot	100	1934.15590	1H
4	4	180055441	2565549	692	395636	Shot	10	Shot	100	289.39887	2H

y1	y2	x1	x2	tags_id1	tags_id2	tags_id3	tags_id4	tags_id5	tags_id6	gol	propiaGol
34.00	21.08	93.45	99.75	101	402	1202	1801	NA	NA	1	0
40.80	0.00	100.80	0.00	101	402	201	1204	1801	NA	1	0
16.32	68.00	75.60	105.00	101	401	201	1204	1801	NA	1	0
25.16	0.00	95.55	0.00	101	402	201	1201	1801	NA	1	0

Figura 33: Extracto del data frame *goals.csv*

Otro data frame se denomina *resumenes.csv*. Este data frame se utiliza para generar las representaciones de las Figuras 9,10, 11 y 12, y se obtiene a partir de *matches.csv*, *teams.csv* y *goals.csv*. Nos servimos del fichero *matches.csv*, y concretamente de su campo *label*, para obtener el resultado final del partido, los nombres de los equipos que lo disputaron y la condición de local/visitante de cada uno. Una vez extraídos los nombres de los equipos, usamos el fichero *teams.csv* para obtener sus identificadores, ayudándonos para ello del campo *name* y obteniendo el identificador a través de *wyId*. Por último con el fichero *goals.csv* definimos para cada partido el número de goles del equipo local y el equipo visitante tanto al descanso como al final del partido, y además en ambos casos damos el resultado en formato "quiniela", indicando con un *1* una victoria local, con un *2* un empate y con un *3* una victoria visitante. En la Figura 34 se muestran las 4 primeras entradas de este nuevo fichero.

	id_partido	id_eqLoc	eqLoc	id_eqVisit	eqVisit	golLoc	golVisit	golesLocHT
1	2565922	676	Barcelona	687	Real Sociedad	1	0	0
2	2565919	678	Athletic Club	691	Espanyol	0	1	0
3	2565927	682	Villarreal	675	Real Madrid	2	2	0
4	2565923	714	Las Palmas	756	Girona	1	2	1
5	2565918	692	Celta de Vigo	695	Levante	4	2	2

golesVisHT	primerGol	periodoPrimerGol	ultimoGol	periodoUltimoGol	quinHT	quin	country
0	676	2H	676	2H	2	1	Spain
1	691	1H	691	1H	3	3	Spain
2	675	1H	682	2H	3	2	Spain
2	756	1H	756	1H	3	3	Spain
1	695	1H	695	2H	1	1	Spain

Figura 34: Extracto del data frame *resumenes.csv*

Tras construir el fichero *resumenes.csv* extraemos los distintos subconjuntos de datos en función del gráfico que deseemos realizar y con la librería *ggplot* creamos la representación gráfica.

C. Anexo III: Generación de variables del modelo

En este Anexo se describen los aspectos de carácter técnico que fueron tenidos en cuenta para generar las variables: *ángulo*, *accionPrevia*, *tiempoJugada*, *distanciaJugada* y *velocidadJugada*. Estos fundamentos no fueron explicados en el Capítulo 5 para no hacer muy densa la lectura.

C.1. Construcción de la variable accionPrevia

Como se mencionó en el Capítulo 5.2.1, para generar la variable *accionPrevia* se empleó el etiquetado del evento anterior al tiro (ver Capítulo A.2), dando lugar a un total de 11 categorías. Las características de cada una de ellas son:

- Acc. Individual: el jugador que realiza el tiro previamente ha regateado a un rival o ha avanzado una distancia considerable de metros con el balón. Se etiquetan bajo esta categoría todos los tiros cuyo subevento anterior estén catalogados como número 11 o 70, en caso de ser protagonizados por el mismo jugador que el tiro, o el número 12 si se trata del equipo rival.
- Acc. Portero Contrario: si el portero rival realiza una salida en falso o el tiro es fruto de un despeje del portero a un tiro anterior. Los subeventos previos de interés están etiquetados como 40, 90 y 91.
- Balón suelto: cuando el tiro viene precedido de un momento de incertidumbre en el que ninguno de los equipos tenía de manera clara la posesión del balón. Hay dos subeventos dentro de esta categoría y son el número 13 y el 72.
- Centro: si el remate efectuado se produce justamente después de un centro o tras encontrarse el balón en el aire. Los subeventos que hacen referencia a un centro son el número 30 y el 32, mientras los referidos a una situación aérea del balón son el número 10, 80 y 82.
- Pase: el tiro viene precedido de un pase sin ninguna peculiaridad asociada. El número del subevento pase es el 85.
- Pase alto: el jugador realiza el disparo después de recibir el balón por medio de un pase por alto. No se debe confundir con la categoría *Centro*. La diferencia entre ambas radica en que en *Pase alto* en el momento del envío del balón hacia el jugador que realiza el tiro no existe un posicionamiento global del equipo o de dicho jugador que invite a pensar que va a ocurrir una acción de gol, mientras que en el caso de *Centro* sí. El subevento asociado a esta categoría está catalogado como el número 83.
- Pase inteligente: el tiro viene precedido de un pase que ha roto las líneas defensivas del equipo rival. Este tipo de pase se etiqueta como el subevento número 86.

- Reanudación juego: comprende todos aquellos lanzamientos a puerta que vienen precedidos de una acción de reanudación, como pueden ser un saque de falta o de banda. Los subeventos de interés son el número 31 y 36.
- Robo: el tiro se realiza inmediatamente después de que el equipo que lo efectúa haya robado la pelota al equipo rival. Los subeventos contenidos en esta categoría son los números 12, 11 y 71.
- Tiro previo: en caso de que exista un primer lanzamiento efectuado justo en el instante de tiempo anterior y que por diversos motivos fue bloqueado antes de llegar a ser interceptado por el portero o acabar fuera del terreno de juego. La principal diferencia respecto a *Acc. Portero Contrario* es que en este caso no hay intervención del portero. Se consideran tanto los tiros a balón corrido como los efectuados desde el punto de penalti o en el lanzamiento de una falta. Los subeventos de interés son el 10, el 33 y el 35.
- Otros: comprende todas aquellas situaciones que no han sido categorizadas en ninguna de las situaciones descritas. Comprende los subeventos 34, 50, 71, 81, 84 y los eventos 4 y 9, siempre que estos sean protagonizados por el mismo equipo que efectúa el lanzamiento.

Con la creación de estas 11 categorías se ha tratado de representar de la mejor manera posible las diferentes situaciones que se pueden dar durante un partido. Como es lógico pensar, es posible que se haya englobado bajo una misma categoría situaciones muy distintas ya que solo se ha considerado el evento inmediatamente anterior al tiro y no el cúmulo de eventos anteriores que permitiría un mejor etiquetado.

C.2. Determinar el inicio de una jugada

Para generar las variables *tiempoJugada*, *distanciaJugada* y *velocidadJugada* ha sido necesario detectar el inicio de la jugada asociada a cada tiro. Para ello, se ha recurrido al fichero *events.csv*.

En el fichero *events.csv* se cuenta con todos los eventos que tuvieron lugar en cada uno de los partidos, por lo que para hallar el inicio de una jugada es suficiente con ir retrocediendo en las observaciones hasta detectar un patrón que asociemos al final de una jugada previa y el comienzo de la actual.

Para determinar el origen de la jugada se debe tener en cuenta el equipo que protagoniza el evento que se está estudiando. Por ello, se establecen diferentes condiciones que marcan el final de una jugada y el comienzo de otra en función de si el evento que se está evaluando lo protagoniza el equipo que termina realizando el tiro o su rival.

En el caso de que el evento a valorar esté protagonizado por el equipo que realiza el tiro y además se trate de una reanudación de juego (evento número 3) entonces este representa el comienzo de la

jugada. Si por el contrario, el evento es una parada (evento número 9) o una falta (evento número 2) realizada por el equipo protagonista, entonces el comienzo de la jugada se prudice en el evento siguiente.

Cuando el evento es protagonizado por el equipo rival se considera que se trata de una jugada distinta siempre y cuando se trate de una falta (evento número 2), una acción de reanudación del juego (evento número 3), una interrupción (evento número 5), un fuera de juego (evento número 6), una parada (evento número 9), un tiro (evento número 10), una aceleración (sub evento número nº 70) o un despeje (subevento número 71). En caso de que el evento no sea ninguno de los mencionados con anterioridad y se trate de un pase (evento número 8), se considerará que se corresponde con el final de la jugada previa si el evento inmediatamente anterior vuelve a estar protagonizado por el mismo equipo y además es una falta, un pase o un duelo de ataque (sub evento número 11).

Tras determinar el evento que da inicio a la jugada, se procede a realizar el cálculo de la variable *tiempoJugada*. Para ello, se resta al valor de la variable *segundos90* asociada al evento del tiro, el valor de esa misma variable para el evento considerado como inicio de la jugada.

Para calcular *distanciaJugada* se recorren todos los eventos acontecidos entre el tiro y el inicio de la jugada, calculándose para cada evento la distancia entre el par (x1,y1) y el par (x2,y2). Como dentro de la secuencia de eventos a recorrer pueden existir algunos protagonizados por el equipo rival, como los duelos aéreos o defensivos, solo se consideran los eventos protagonizados por el equipo que realiza el tiro. Para evitar que se pierda información sobre la distancia recorrida por el balón al excluir algunos de esos eventos, además de calcular la distancia recorrida como consecuencia del evento en cuestión, se calcula la distancia existente entre la posición final del balón en el evento *i* y la posición inicial en el evento *i+1*.

C.3. Cálculo del ángulo de tiro

Para calcular el ángulo de tiro desde una posición *(x,y)* se ha empleado la formulación realizada por César A. Morales en su estudio *"A mathematics-based new penalty area in football: tackling diving"* [28]. En este trabajo se propone el uso de un nuevo tipo de área en las porterías ya que considera que las actuales contemplan zonas del terreno de juego que no representan una posición peligrosa para el equipo defensor. Por ello, basándose en el ángulo respecto a la portería define un nuevo tipo de área que únicamente incluye las zonas con buenos ángulos respecto a la portería.

Para el cálculo del ángulo considera los siguientes valores:

- *a* y *c*: distancia en metros desde la posición del balón a cada uno de los postes de la portería.
- *d*: distancia desde la posición del balón al centro de la portería.
- *b*: longitud de la portería. En nuestro caso, 7.32 metros.

- γ: ángulo formado por la línea de gol y cada una de las líneas imaginaria que unen la posición del balón con el poste de la portería.

Por tanto, la disposición de todos los valores se muestra en la Figura 35.

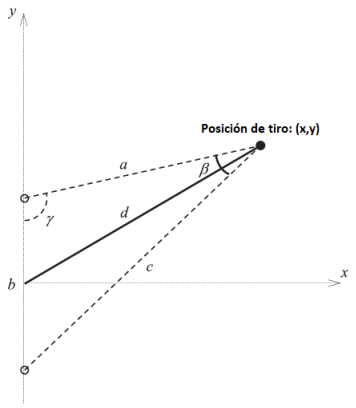

Figura 35: Situación propuesta por César A. Morales.

La relación propuesta por Morales [28] entre el **ángulo de tiro** β y los componentes geométricos de la Figura 35 es la establecida en la Ecuación 10.

$$\beta = \arctan(\frac{bx}{x^2 + y^2 - (\frac{b}{2})^2}) \tag{10}$$

Los valores de x e y en nuestro conjunto de están medidos tomando como origen de coordenadas el centro de la portería del equipo que protagoniza el evento. Por el contrario, en la modelización propuesta para la Ecuación 10 el origen de coordenadas es la portería del equipo contrario, es decir, sobre la que se realiza el tiro. Por ello, se ha de tener en cuenta que la coordenada x de nuestros eventos no se corresponde con la x de la Ecuación 10, sino que $x_{modelo} = longitud\_campo - x_{datos}$. Es necesario realizar la adecuación de la coordenada x según la relación anterior antes de calcular el ángulo. El valor de *longitud_campo* depende de las dimensiones consideradas para el terreno de juego, en nuestro caso es 105 metros.

Bibliografía

[1] GrupoBit, "¿Cuántos datos se producen en un minuto?." https://business-intelligence.grupobit.net/blog/cuantos-datos-se-producen-en-un-minuto.

[2] E. C. Digital, "La vuelta del fútbol, el negocio que genera el 1,37 por ciento del PIB y más 185.000 empleos." https://elcierredigital.com/pizarrra-deportiva/42459131/vuelta-futbol-genera-pib-empleo.html.

[3] L. Pappalardo and E. Massucco, "Soccer match event dataset," Feb 2019.

[4] J. N. Morgan and J. A. Sonquist, "Problems in the analysis of survey data, and a proposal," *Journal of the American Statistical Association*, vol. 58, no. 302, pp. 415–434, 1963.

[5] C. J. A. González, "Introducción al aprendizaje de ensembles," *Minería de Datos. Departamento de Informática. Universidad de Valladolid.*, 2020.

[6] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer New York, 2013.

[8] C. Lemaréchal, "Cauchy and the gradient method," 2012.

[9] D. Rodríguez, "Implementación del método del descenso del gradiente en python." https://www.analyticslane.com/2018/12/21/implementacion-del-metodo-descenso-del-gradiente-en-python/, 2018.

[10] C. Li, "A gentle introduction to gradient boosting." http://www.chengli.io/tutorials/gradient_boosting.pdf.

[11] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 – 1232, 2001.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," pp. 785–794, 08 2016.

[13] A. Jain, "Complete guide to parameter tuning in xgboost with codes in python." https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/, 2016.

[14] J. Ooms, "The jsonlite package: A practical and consistent mapping between json data and r objects," *arXiv:1403.2805 [stat.CO]*, 2014.

[15] H. Wickham, R. François, L. Henry, and K. Müller, *dplyr: A Grammar of Data Manipulation*, 2021. R package version 1.0.6.

[16] "Glosario de términos wyscout." https://dataglossary.wyscout.com/.

[17] V. Barnett and S. Hilditch, "The effect of an artificial pitch surface on home team performance in football (soccer)," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 156, no. 1, pp. 39–50, 1993.

[18] J. Ensum, R. Pollard, and S. Taylor, "Applications of logistic regression to shots at goal in association football: calculation of shot probabilities, quantification of factors and player/team," *Journal of Sports Sciences*, vol. 22, no. 6, p. 504, 2004.

[19] A. Ryder, "Shot quality." `http://hockeyanalytics.com/Research_files/Shot_Quality.pdf`, January 2004.

[20] H. Hamilton, "Moneyball and soccer." `https://www.soccermetrics.net/high-level-discussions/moneyball-and-soccer-2`, 1999.

[21] S. Green, "Assesing the performance of premier league goalscorers." `https://www.statsperform.com/resource/assessing-the-performance-of-premier-league-goalscorers/`, 2012.

[22] "Portal de datos fbref." `https://fbref.com/es/?lang=es`.

[23] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, and Y. Li, *xgboost: Extreme Gradient Boosting*, 2021. R package version 1.4.1.1.

[24] L. Henry and H. Wickham, *purrr: Functional Programming Tools*, 2020. R package version 0.3.4.

[25] A. South, *rnaturalearth: World Map Data from Natural Earth*, 2017. R package version 0.1.0.

[26] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[27] FIFA, "Convocatorias de los combinados nacionales para el mundial de rusia 2018." `https://img.fifa.com/image/upload/hzfqyndmnqazczvc5xdb.pdf`.

[28] C. A. Morales, "A mathematics-based new penalty area in football: tackling diving," *Journal of Sports Sciences*, 2016.