Proyecto de Datos I

Despliegue y Evaluación del Modelo

Carlos Mantilla Mateos

Álvaro Enol Alonso Ortega

1. Planificación del Despliegue

Nuestro modelo tiene como propósito predecir el valor de mercado de los jugadores en función de sus estadísticas de años anteriores y de distintas variables. Las estadísticas de un partido y de un jugador a lo largo de sus temporadas no necesariamente tienen que ser extraídas en tiempo real, ya que al final de un partido se pueden almacenar, por lo que haremos predicciones de tipo batch en vez de online.

Integraremos todo a través de dos bases de datos: una que contenga las estadísticas y demás variables que hagan referencia al jugador junto con su ID, que servirá como referencia a otra base de datos donde se contenga la predicción del valor de mercado del jugador. También podríamos utilizar APIs, pero lo consideramos menos conveniente, ya que esta información no será pública, sino que será vendida a empresas o clubes.

Los momentos de mayor consulta serían en verano o invierno, cuando se abren las ventanas de fichajes y los clubes pueden solicitar acceso a la base de datos para obtener el valor de los jugadores. Aunque durante la temporada regular no se puedan fichar jugadores, los clubes pueden tener el valor de los jugadores ya previsto para ficharlos cuando comience la ventana de fichajes.

En general, no creo que nuestra base de datos tenga problemas de peticiones, ya que no está abierta al público, sino que trabajamos solo con empresas y clubes. Por lo tanto, no tendrá tanta demanda como si fuera pública.

Sin embargo, habrá que escalar la base de datos para prevenir las cargas pico.

2. Reentrenamiento del Modelo

Para esta fase, tuvimos que hacer actualizar la separación de train y test entre otras, el archivo que antes separaba entre train validación y test, simplemente lo tomamos como train, haciendo a todo esto un fit_transform, y además, aparte de devolver X_train, X_test, RANDOM_STATE, ahora devuelve también la variable scaler del StandardScaler utilizado para hacer el fit_transform con todos los datos de train, por lo que para el archivo de test, tendremos en cuenta este escalado que ha aprendido, para hacer el transform en test.

Sin embargo, cuando recogimos nuevos datos, hay variables que se quedan obsoletas, ya que difieren con respecto a nuevos datos, ya que puede ser que en un año haya por ejemplo unas nacionalidades o equipos y en otras otro, por lo que lo que hemos hecho ha sido quedarnos con las comunes.

Una vez hecho todo esto, la fase que queda es el propio fit y la evaluación del modelo.

3. Captura de Nuevos Datos

Hemos seleccionado como nuevos datos, los de la temporada 23/24, que son los más recientes, en general el proceso de captura ha sido fácil, pues teníamos todo documentado correctamente y simplemente ha sido correr el código con los nuevos enlaces. Se cogieron las estadísticas y precios de las páginas hasta la jornada número 33 (la más reciente a fecha de captura).

4. Monitorización de la Deriva

Para esta fase, en la elección de las 3 variables más relevantes, no solo se ha tenido en cuenta la importacia de las variables en el modelo, sino la adaptación a datos nuevos, para empezar las variables de equipos y nacionalidades no las vamos a tener en cuenta para esta elección, porque pueden ser variables que luego no existan para otros modelos o que si decidimos expandir el modelo a otros mercados no nos servirán, por esta razón las quitamos de esta elección, aunque sirvieran a continuación tenemos otra explicación.

El siguiente punto es tener en cuenta variables que no solo sean muy buenas para predecir un tipo de modelos, ejemplo, variables muy buenas para predecir variables de un tipo de posición, por ejemplo goles o asistencias, que son muy buenas, pero que no sirven para algunos conjuntos, que pueden quedar marrginados.

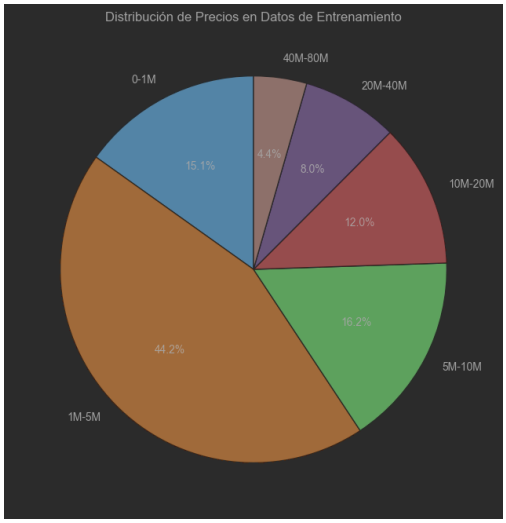
Por lo que teniendo en cuenta esto y la importancia en sí de las variables hemos decidido coger las siguientes, la primera es *1_año_anterior*, que nos sirve para tener un precio inicial con el que ajustar, además el modelo la marca como muy importante, la siguiente va a ser *mins*, una variable clave para cada sector, porque sirve para todos los conjuntos diferentes de jugadores, es la variable más correlacionada con la respuesta, y además se marca como muy importante. Para la última elección, hemos decidido coger *nationality_germany*, una de las variables que se marcaba como importantes del modelo, podríamos haber cogido otra importante del estilo de *AvgP*, pero decidimos analizar una categórica para tener otra visión de variables diferentes.

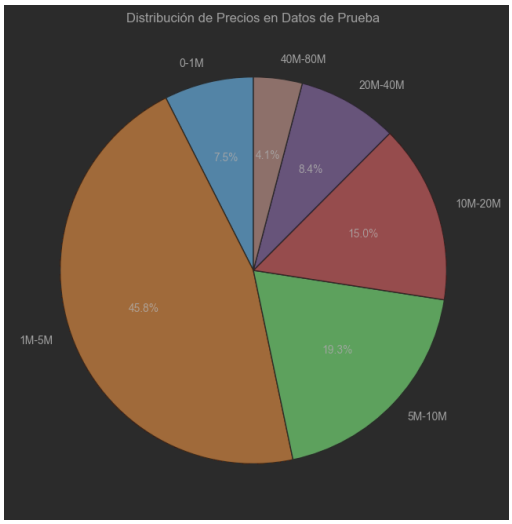
Análisis de la variable Respuesta

Se ha hecho un análisis a través de gráficas, para ver cambios en las tendencias de la variable respuesta, procederemos a verlos y a comentar posibles cambios, tras realizar el análisis, podemos ver que el precio con el pasar de los años va aumentando, viendo para los datos de entrenamiento de 0 a 1 millón, casi el doble de jugadores que en test proporcionalmente.

Por lo que podemos ver una inflada de precio en esta franja, que quiere decir que ya no hay casi jugadores de menos de un millón, los que antes representaban una parte significativa de los jugadores, esta diferencia se ha repartido en otras franjas de 1 millon a 20 aproximadamente, donde vemos un aumento significativo en los datos de test (temporada 23/24).

Sin embargo, algo curioso, es que las super estrellas (más de 20 millones), siguen representando aproximadamente el mismo porcentaje, lo que puede significar que estos jugadores caros no se han ido de la liga, o que cuando se van se contratan otros del mismo precio aproximadamente.

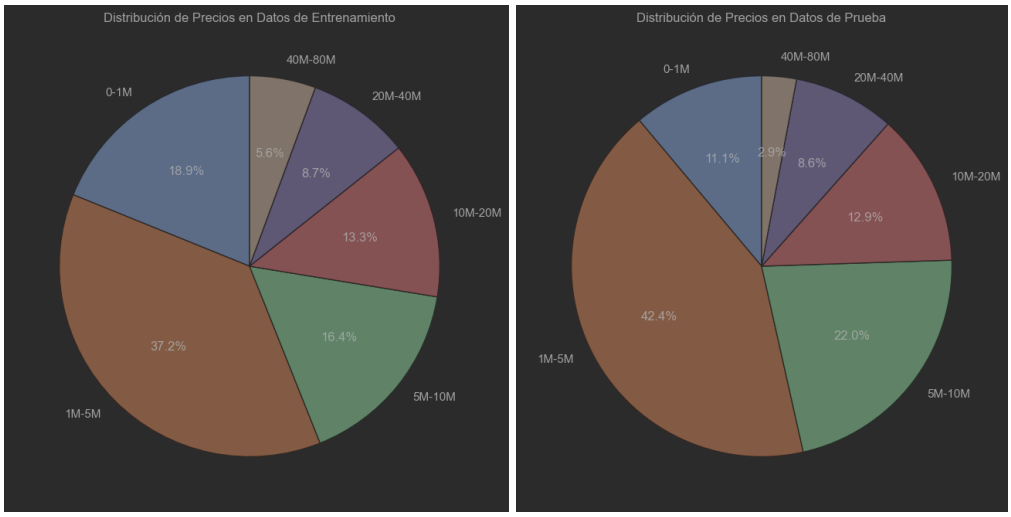




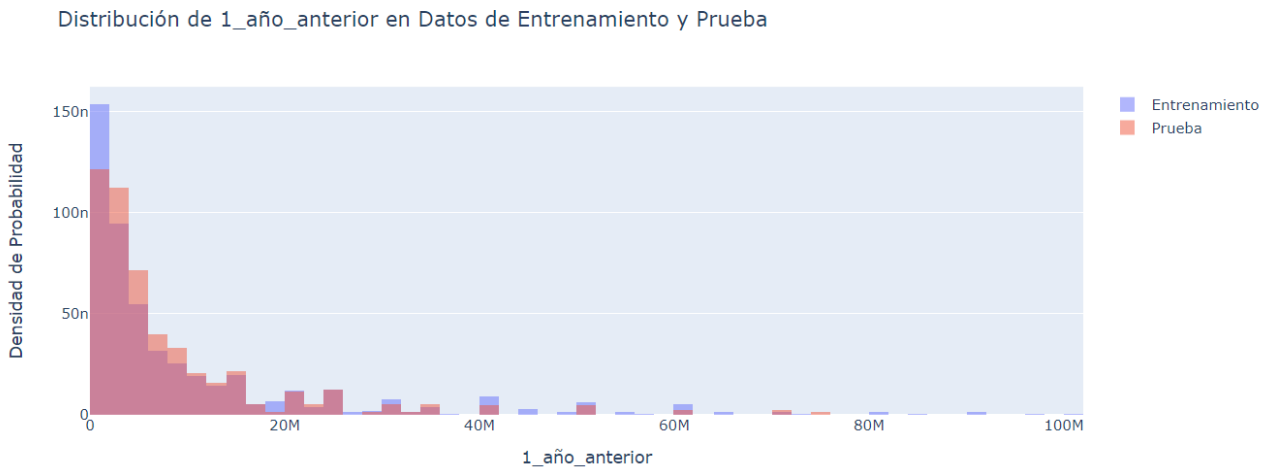
Análisis de 1_año_anterior

En este caso, observamos un comportamiento similar al anterior, aunque un poco más gradual, pues vemos como se reduce el número de juegadores de menos de un millón, y cada vez se reduce más, esa bajada significativa de un 5 % (de los años anteriores al 22/23), que sigue bajando en la temporada actual bruscamente, otro 4 %, de un año para otro, esta tendencia se puede deber a la competitividad en el sector, pues viendo la popularidad mundial de la liga española hoy día, cada vez más gente quiere pertenecer a ella, al haber más gente, hay que entrenar más para poder llegar, por lo que cada vez los jugadores que debutan son más buenos y tienen mejores estadísticas.

Nota: En la segunda gráfica se puede ver claro también, cada barra representa 2 millones, esos despuntes del conjunto de entrenamiento en precios de 2 a 20 millones.

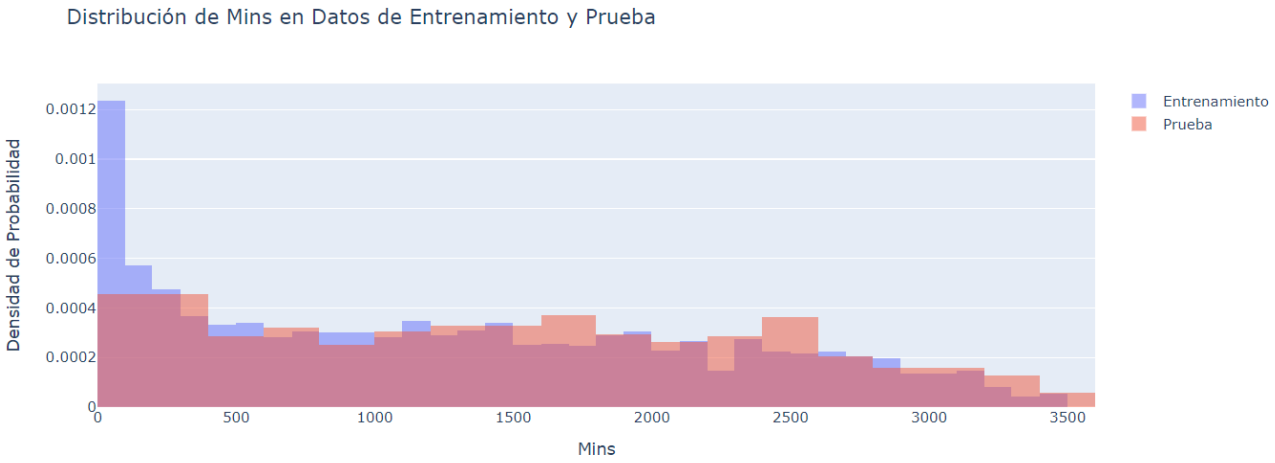


Nota: Para las siguientes gráficas, se ha hecho una proporción para poder comparar los datos, puesto que los de entrenamiento tienen muchos más registros, y no sería posible comparar a simple vista.



Análisis de Mins

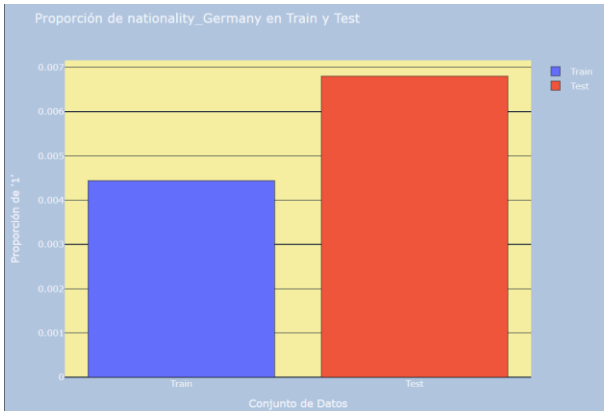
Con respecto a la variable minutos, podemos ver como en años anteriores hay un pico de jugadores que no jugan casi ningún partido, que se reduce a más de la mitad en esta última temporada, a partir de este punto predomina la temporada 23/24 en prácticamente todas las zonas a partir de 400 minutos, incluso llegando a un número de minutos nunca antes vistos, que superan los 3500 minutos, llega a los 3599, las 38 jornadas sin contar con descuentos suman un total de 3420 minutos.

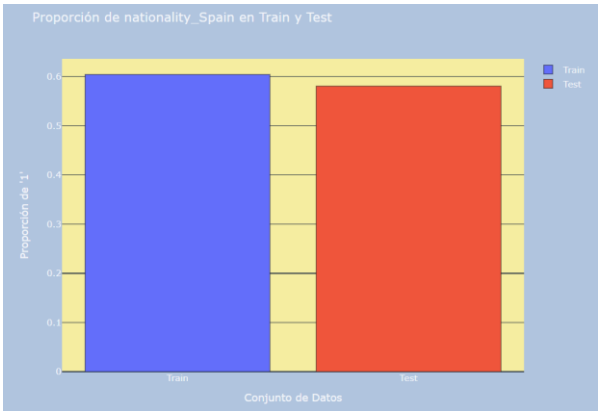


Análisis de nationality_germany

Esta variable se utilizó para no caer en la monotonía de variables numéricas, para empezar, esta variable se marca como la variable dentro de nacionalidades más importante para predecir el modelo, por lo que vimos interesante analizarla. Las principales son que el número de jugadores en la liga entre entrenamiento y test, ha duplicado prácticamente, lo que puede decir que es un mercado en el que sus jugadores no defraudan.

También quisimos ver la nacionalidad_española, pues hay últimamente muchas noticias de que las liga española importa muchos jugadores, y casi no exportan jugadores locales, por lo que resultó interesante también ver esta, el número de jugadores españoles se ha mantenido en test, lo que puede decir que apreciamos nuestro talento local, y que al fin y al cabo, al tener una de las ligas más importantes del mundo, nuestros jugadores no ven la necesidad de tener que irse.





5. Análisis de Resultados

Hemos encontrado un problema a la hora de evaluar el modelo de regresión lineal, y es que a la hora de recuperar nuevos datos, la columna OffDrb, no se encontraba ya en el conjunto debido a una actualización de la página web, por lo que evaluamos la regresión lineal sin esta variable, y pasaba de un error de 2,567,496.50 dato sacado de la entrega anterior, a 3,419,807.08, por lo que notamos que empeora bastante el modelo.

Decidimos volver a evaluar la selección de variables, para ver si incluía otra que pudiéramos utilizar por esta, pero de nuevo el modelo con kbest nos marcaba esta vez el mejor error con 19 variables, que son las mismas que incluía antes pero sin OffDrb, pero pasa de ese 2,567,496.50 a 2,658,272.75 de la entrega anterior, que se refleja en el mismo valor nuevo 3,419,807.08, por lo que no tuvimos otra que regresar a los modelos anteriores, para comparar evaluaciones a ver si alguna de estas solucionaba esta pérdida de rendimiento.

Probamos con la red neuronal, que era el segundo mejor modelo que nos daba, comparando, para ver si había una mejora de rendimiento, resultó en que, primero que nada retrocedimos al modelo previo, para hacerle unas cuantas pruebas nuevas de hiperparámetros, poniendo todo el foco en este modelo, para sacar los mejores parámtros posibles, esta vez hicimos una búsqueda amplia como la anterior en todos los parámetros, excepto en hidden_layer_sizes, que dada su importancia la exploramos a fondo, probando el modelo de 1 a 200 neuronas en cada capa, para un número de capas del 1 al 3, que anteriormente solo probamos con una capa, y sorprendentemente, ¡Hemos conseguido superar a la regresión linal¡

Consiguiendo un error de 2,447,816.31, que mejora con respecto al 2,629,609.35 que salía anteriormente, esta vez los hiperparámetros utilizados son los siguientes: solver = adam, learning_rate_init = 0.0001, hidden_layer_sizes = (100, 170, 10), batch_size = 64 y alpha = 0.1. Ahora vamos a ver qué tal funciona nuestro modelo con nuevos datos, al probarlo observamos un nuevo valor de 2,638,345.93, con el resultado de que empeora, pero no una locura, y aún así sale mucho mejor que la regresión lineal que teníamos antes, por lo que este cambio ha resultado ser un completo éxito.

En cuanto al rendimiento separado por precios, podemos ver los siguientes resultados, antes de los nuevos datos daba de 0 a 1 millon 717,665.11, de 1 a 5 millones 1,119,523.07, de 5 a 10 millones 2,043,061.76, de 10 a 20 3,109,321.64, de 20 a 40 millones 6,616,392.51 y de 40 a 80 millones, 16,091,888.99.

Mientras que ahora con la captura de nuevos datos da lo siguiente, de 0 a 1 millon 789,741.14, de 1 a 5 millones 1,290,622.87, de 5 a 10 millones 1,820,011.14, de 10 a 20 3,584,527.86, de 20 a 40 millones 5,200,632.31 y de 40 a 80 millones, 16,021,540.02. Mejora para valores de 5 a 10 y de 20 a 40, lo cual tiene sentido con ese mercado al alza del que hablábamos anteriormente.

En el rendimiento separado por posiciones, podemos ver los siguientes resultados, antes de los nuevos datos daba, en defensas 2160490.15, en delanteros 2,268,857.05, en medios 3,216,548.15, y en porteros 2,145,369.86.

Mientras que con la nueva captura de datos, da, en defensas 2,376,539.17, en delanteros 2,041,971.35, en medios 3,699,030.38, y en porteros 2,252,638.36. No hay nada muy notable que destacar aquí, vemos modificaciones en algunos precios, sobre todo, la diferencia más grande está en mediocampistas, puede ser que este alza de jugadores esté relacionado con que muchos de ellos son mediocampistas.

Por su parte, la heurística, que se trataba de aplicar un factor de apreciación si tiene menos de 30 años, o depreciación si los supera, esta edad se estableció según diferentes informes del FC Barcelona y La Liga acerca del rendimiento de los jugadores, ambos concuerdan que los jugadores mayores de 30 años empiezan a perder facultades, por lo que menos habilidades, menos estadística, se relaciona directamente con la bajada de precio también, tras aplicarla nos da un resultado bastante malo de: 8,703,514.75, este rendimiento se debe a que las relaciones de este tipo de datos son más complejas que una simple heurística.

Las heurísticas están bien planteadas en algunos proyectos, porque igual estás prediciendo la temperatura que hay mañana por ejemplo, vienes trabajando un modelo super complejo de red neuronal con no se cuantas capas, y resulta que alguien que dice que la temperatura que va a haber mañana es la de hoy y acierta más que tú, es un cálculo rápido que te permite saber si realmente tu modelo merece la pena desplegarlo o no, en nuestro caso si.