Proyecto de Datos I

Entrenamiento de datos

Óscar Marín Esteban

Carlos Mantilla Mateos

Álvaro Enol Alonso Ortega

1. Definir datos y estrategia de validación

Selección de datos:

Para esta fase de entrenamiento, hemos descartado datos de jugadores cuyo valor de mercado sea menor a 80 millones. Nos quita solo 16 valores, de 2267 a 2251. Esta decisión mejora la calidad del modelo, ya que solo esos valores nos estaban dando casi 1 millón más de error.

Para empezar, consideraremos todas las variables del conjunto de datos, excepto una que se llama index, que se puso en la anterior fase para facilitar el junte de algunos datos, pero luego para el entrenamiento de los modelos, para el caso de regresión lineal utilizaremos distintos métodos de selección de variables, para el resto de los casos, se escogerá según la conveniencia de cada algoritmo automáticamente.

Definición de estrategia de validación:

Utilizamos validación cruzada (cross-validation) con k = 10 en todos los casos. Esto implica dividir el conjunto de datos en 10 partes iguales, utilizando 9 partes para el entrenamiento del modelo y 1 parte para la evaluación. Este proceso se repite 10 veces, de manera que cada parte se utilice una vez como conjunto de prueba.

Hemos inicializado la semilla a 83 para garantizar la consistencia en los resultados.

Métricas de rendimiento:

Hemos decidido utilizar el error medio absoluto como métrica de rendimiento para evaluar los modelos. Esta métrica es más consistente frente a valores atípicos que el error cuadrático medio, lo cual es relevante dado el tamaño reducido del conjunto de datos.

Al principio consideramos el error cuadrático medio, pero no se adaptaba nada bien.

Para terminar con este apartado, queríamos mencionar que se probó el utilizar solo los datos del año 2022 para la predicción, pero daba peores resultados, que considerando todo.

2. Transformación de los datos

Transformaciones de variables categóricas:

Para nuestras variables categóricas, que son *Nacionalidad*, *Equipo* y *Posición,* habíamos usado el método de LabelEncoder, pero resultó en un funcionamiento que no esperábamos, ya que este algoritmo interpreta los valores como numéricos reales.

Como solución, optamos por utilizar el método OneHotEncoder, el cual convierte las variables categóricas en variables dummy, asignando un valor binario a cada categoría. Esta es la transformación que se hará al modelo, pero es común para todos.

Escalado de datos:

Como algunos datos tienen rangos dispersos, decidimos aplicar escalado a las variables. El escalado de datos ayuda a que las variables se adapten a un rango más reducido.

3. Análisis de variables relevantes

Basándonos tanto en la fase de exploración en la correlación con la respuesta, como en algunos algoritmos de selección de variables como el mrmr, tenemos algunos candidatos a elegir como las más relevantes, sin embargo no hemos podido concluir ninguna de las variables como no relevante o de no uso, algunas de las más importantes son: *mins, goals, asssists, MotM, Inter, KeyP, AvgP, 1_año_anterior, 2_año_anterior, 3_año_anterior, 4_año_anterior.*

Se compararán con las que se cojan luego en el modelo de regresión lineal.

4. Entrenamiento del modelo

Regresión Lineal

Selección de Variables:

La selección de variables es relevante para el modelo de regresión lineal, ya que nos permite generalizar mejor y asignar pesos para cada variable.

Inicialmente, probamos el algoritmo mRMR (Minimum Redundancy Maximum Relevance), que combina la selección de variables para predecir la variable objetivo con poca redundancia entre ellas.

Tras varias pruebas seleccionando iterativamente cada vez una variable más, siguiendo una secuencia hacia delante, nos resulta que, en general el rendimiento entre más número de variables es similar conforme van pasando a utilizar más número de variables, sin embargo en el número 17 de variables pega una bajada significativa de aproximadamente 100k, pasa de aproximadamente 3032067.7841, a 2956767.2838, contando con las siguientes variables: *'1_año_anterior', 'nationality_Denmark','nationality_Poland','nationality_Costa Rica','nationality_Mexico','nationality_Guinea','nationality_Zambia','nationality_Greece','nationality_Montenegro','nationality_Nigeria','nationality_Slovakia', 'Año_natural', 'nationality_England', 'nationality_Armenia', 'nationality_Norway', 'nationality_Dominican Republic', '2_año_anterior'*, aquí podemos ver como el 1 año anterior(valor al principio de la temporada), 2 año anterior o el año_natural, tienen sentido que sean de las variables más importantes a la hora de generar el modelo, sin embargo echamos en falta otras importantes como pueden ser los minutos o los goles, por poner dos ejemplos.

A partir de este punto, la única bajada considerable es con 71 variables, que se queda en un: 2907383.7577, pero creemos que tampoco es tan significativa para todo el número de variables que incluye de más, ya que se prefiere sacrificar un mínimo de rendimiento por simplicidad, a partir de las 140 variables, el error aumenta muy considerablemente.

Posteriormente, utilizamos otra técnica de selección, en este caso la técnica kBest, que selecciona las k mejores características basadas en pruebas estadísticas. Encontramos que el menor error se obtiene con 9 variables, incluyendo: *'Mins' 'Assists' 'MotM' 'Titularidades' 'Equipo_pos' '1_año_anterior', '2_año_anterior' '3_año_anterior' '4_año_anterior',* estas variables si nos cuadran un poco más con que sean las importantes, es más, la mayoría concuerdan con las que habíamos escogido a priori.

Adicionalmente, un apunte que queremos mencionar, es que con 17 variables, número que casualmente coincide con el punto anterior de mrmr, aparte de ser el segundo mejor punto y coger variables parecidas a las que coge con 9, hace cierta una hipótesis que habíamos hablado de que, que un jugador sea de los equipos más grandes (FC Barcelona, Real Madrid), se tiene bastante en cuenta a la hora de predecir, este modelo lo demuestra, al coger las siguientes variables: *'Mins' 'Goals' 'Assists' 'SpG' 'MotM' 'KeyP' 'OffDrb' 'AvgP', 'Titularidades','Equipo_pos','1_año_anterior','2_año_anterior','3_año_anterior','4_año_anterior','5_año_anterior' 'Equipo_FC Barcelona', 'Equipo_Real Madrid'.*

Con este análisis damos por concluida la fase de selección de variables para el modelo de regresión lineal.

Ajuste de Hiperparámetros:

Dado el enfoque en la selección de variables, no tiene sentido realizar el ajuste de hiperparámetros para el modelo de regresión lineal.

Evaluación del modelo:

Se han realizado dos pruebas para comparar el rendimiento entre dos modelos utilizando diferentes conjuntos de variables. En el primer caso, se usaron las 9 variables seleccionadas en la fase anterior, mientras que en el segundo caso se utilizaron todas. Los resultados son los siguientes, el modelo con las 9 variables obtuvo un MAE de 2,787,930.65 en el conjunto de prueba, mientras que el modelo con todas las variables dio un MAE de 3,015,336.51. Esta mejora en el rendimiento concluye en que la selección de variables ha sido positiva y que algunas de las variables incluidas previamente podrían haber sido irrelevantes o incluso perjudiciales para el modelo.

Además, hemos utilizado el coeficiente de determinación ajustado (R^2 ajustado) como medida de evaluación adicional del rendimiento del modelo. Esta elección se debe a que el R^2, tiende a aumentar artificialmente con la inclusión de más variables independientes. El R^2 ajustado aborda esta limitación al penalizar el número de predictores en el modelo, proporcionando así una medida más precisa del ajuste del modelo a los datos. Da un R^2 ajustado de 0.82.

Aunque se realizó un análisis por segmentos separando los datos por posición, este análisis solo se incluyó en el modelo de árbol. Estos resultados los veremos a continuación, mostrando el por qué no se han seleccionado finalmente.

Cabe destacar que, para este modelo, el escalado de los datos no fue conveniente, pues subía mucho el error.

Árboles

Selección de Variables:

No tiene sentido usar la selección de variables para este modelo, ya que las selecciona el algoritmo de la mejor manera.

Ajuste de Hiperparámetros:

Para el ajuste de hiperparámetros de este modelo, hemos ido modificando los parámetros según daban mejores resultados, y adaptándolos para seguir esas rutas que daban mejores resultados, los mejores resultados que hemos obtenido son con gridSearch, en el que se pudo bajar aproximadamente 100k con respecto a utilizar los parámetros obtenidos con RandomizeSearch, con gridSearch da un MAE de 2899254.41, con RandomizedSearch da 2981259.77 y sin parámetros da 3354877.04. Los parámetros que vamos a utilizar para evaluar el modelo son: criterion="absolute_error", max_depth = 30, max_features=None, max_leaf_nodes= 80, min_samples_leaf=5, min_samples_split=12, splitter="random".

Evaluación del modelo:

Para el caso de los árboles, se ha optado por probar separar por posiciones para ver si convenía esta separación o no, se cogió bajo las mismas condiciones, el modelo general, y se separó en posiciones según predecían cada jugador, se hizo la media, y comparado con la separación del modelo y la posterior evaluación, resultó más conveniente hacer el modelo todo junto, por lo que a parte de simplificar el trabajo significa que puede ser que tenga en cuenta el completo variables importantes que ayuden a la separación que el específico por posición no tenga en cuenta.

Con respecto al comportamiento del modelo en sí, con los hiperparámetros da un MAE de 3174704.14 y un R^2 ajustado de 0.65, mientras que sin hiperparámetros da 3293565.08, con un R^2 ajustado de 0.56.

Red Neuronal

Selección de Variables:

No tiene sentido usar la selección de variables para este modelo, ya que las selecciona el algoritmo de la mejor manera.

Ajuste de Hiperparámetros:

Para el ajuste de hiperparámetros de este modelo, hemos ido modificando los parámetros según daban mejores resultados, y adaptándolos para seguir esas rutas que daban mejores resultados, los mejores resultados que hemos obtenido son con gridSearch, en el que se pudo bajar aproximadamente 100k con respecto a utilizar los parámetros obtenidos con RandomizeSearch, con gridSearch da un MAE de 2899254.41, con RandomizedSearch da 2981259.77 y sin parámetros da 3354877.04. Los parámetros que vamos a utilizar para evaluar el modelo son: criterion="absolute_error", max_depth = 30, max_features=None, max_leaf_nodes= 80, min_samples_leaf=5, min_samples_split=12, splitter="random".

Evaluación del modelo:

Con respecto al comportamiento del modelo en sí, con los hiperparámetros da un MAE de 3174704.14 y un r cuadrado ajustado de 0.65, mientras que sin hiperparámetros da 3293565.08, con un R^2 ajustado de 0.56.

Cabe destacar que, para este modelo, el escalado de los datos no fue conveniente, pues subía mucho el error.

Como podemos ver, los resultados del R^2 ajustado tanto del árbol como de la red neuronal dejan mucho que desear.

5. Conclusiones

Vamos a optar por el modelo de regresión lineal debido a su simplicidad, es decir, la facilidad de comprenderlo y explicarlo. El modelo de regresión lineal es fácil de entender tanto para personas no técnicas como para expertos en el campo, ya que se basa en relaciones lineales entre las variables predictoras y la variable objetivo.

Además, hemos observado que el modelo de regresión lineal es el modelo que presenta el mejor rendimiento en el conjunto de prueba. Esto significa que el modelo es capaz de hacer predicciones más precisas sobre los precios de los jugadores en datos que no ha visto antes.

Por último, el modelo de regresión lineal tiene un R^2 ajustado más alto en comparación con otros modelos, esto quiere decir, que el modelo es capaz de explicar una mayor proporción de la variabilidad en los precios de los jugadores, lo cual es deseable al construir un modelo predictivo robusto.