Proyecto de Datos I

Entrenamiento de datos

Óscar Marín Esteban

Carlos Mantilla Mateos

Álvaro Enol Alonso Ortega

Alegaciones

1. En la selección de datos debe quedar claro si estáis entrenando con los jugadores de distintas temporadas mezclados. Asumo que sí por el número, pero de qué temporadas. Tampoco queda claro el número de variables, ni las variables descartadas y las razones.

La separación del conjunto de datos se hará según la variable Año_natural, que luego eliminaremos, la diferencia con la anterior entrega es que en este caso en vez de una estrategia de validación con k-folds, haremos entrenamiento con los años 2019 y 2020, validación con el año 2021, y test con 2022, lo que nos dará una visión más realista del funcionamiento de nuestro modelo para próximas temporadas.

El número de variables son 150, ya que al hacer one hot encoding de la variable posición, equipo y nacionalidad pasan de 42 a 150 variables. Las únicas variables que descartamos son Año que contenía la temporada ejemplo (2022-2023) y player que es el nombre del jugador.

2. En la estrategia de validación no estáis considerando partir en 3, sino que solamente hacéis una validación cruzada… es legítimo hacerlo así. Podría tener sentido estratificar la muestra por posiciones de los futbolistas quizá.

Esto lo hemos modificado ya no hacemos una validación cruzada, sino que separamos en tres conjuntos train, validation y test, como hemos mencionado antes.

3. Creo que habiendo algún outlier de precio, es acertado intentar minimizar el EAM para que los errores grandes no tengan tanto efecto.

4. En la memoria no informáis del escalado que lleváis a cabo… Y luego decís que no da buenos, resultados. Esto es extraño. Debéis ver bien qué sucede.

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. Esto lo hemos hecho con StandardScaler().

En esta entrega hemos implementado correctamente el escalado, lo que mejora el rendimiento de nuestros modelos. Los resultados se pueden encontrar en la memoria. Hemos usado datos escalados y no escalados para realizar la comparativa en todos los modelos.

5. Estaría bien documentar en la memoria qué variables a priori tienen más utilidad.

Basándonos tanto en la fase de exploración en la correlación con la respuesta, 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 a priori deberían ser: *mins, goals, asssists, MotM, Inter, KeyP, AvgP, 1_año_anterior, 2_año_anterior, 3_año_anterior, 4_año_anterior.*

6. En el modelo lineal, debéis comentar con kBest el cambio en el error de 9 a 17 variables.

Como hemos comentado anteriormente, al dividir los datos en 3 conjuntos, utilizando el kbest se obtiene que el minimor error se obtiene con 20 variables, un error de 2,653,829.52, y el segundo mejor con 17, un error de 2,661,513.86.

7. No tiene sentido hacer ajuste de hiperparámetros, porque es una regresión lineal y no tiene hiperparámetros que ajustar.

8. Es un poco raro lo de que no funcionen los datos con escalado ==> Debéis revisar por qué sucede esto… Ojo, que solamente debéis escalar las variables cuantitativas. El escalado además es útil porque os permite ver por el "tamaño" del coeficiente, la importancia de las variables y el signo que tienen en la relación.

En la pasada entrega habíamos realizado el escalado sobre todos los datos, en cambio, en esta reentrega lo hemos hecho solo sobre las variables cuantitativas, es era el error por el cual nos empeoraban los modelos escalados, pero ahora podemos afirmar que los 3 modelos mejoran escalando los datos.

9. Cuando usas predicciones con un modelo no lineal no es buena idea usar el R2. Podéis leer más aquí: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2892436/

Hemos eliminado el uso de esta métrica para la evaluación del modelo

10. En el árbol, no decís si escaláis o no los datos. No tiene sentido hacerlo porque el árbol es insensible, pero no lo decís.

11. En el árbol, es extraño que los árboles os den peor que la regresión linea…. Quizá sea un problema de la optimización de hiperparámetros o algún otro problema. Debéis auditarlo bien.

Nos sigue dando que la regresión lineal tiene un error menor que los árboles

12. En el árbol, justo en el árbol tiene menos sentido el separar por posiciones porque un árbol preguntará por las posiciones y hará sub-árboles por posiciones si lo consider adecuado.

Hemos decido realizar la separación por posiciones en la regresión lineal en vez de en los árboles de decisión.

13. En la red neuronal, no decís si escaláis o no los datos.

Comentado en la alegación 4.

14. En la red neuronal, los hiperparámetros que comentáis son los del árbol de decisión.

Modificado.

15. En la red neuronal, es muy extraño que una red neuronal no sea capaz de igualar el modelo de regresión linea. De nuevo, puede ser cuestión de probar una arquitectura adecuada (me parece ver que solamente usáis una capa intermedia) y unos hiperparámetros adecuados o de que haya errores.

Despúes del reentrenamirento, el reajuste de los hiperparámetros y el escalado de los datos esto no ha cambiado.

16. Falta el análisis de las variables importantes y el análisis del error segmentado por posición y/o también por tramos de precio.

Está hecho en la regresión lineal por posiciones y por tramos de precio.

Mencionar que hemos cambiado cosas en toda la práctica en general.

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 y la variable Año_natural, para evitar el data leakage, que hemos usado para la separación de datos pero que posteriormente hemos eliminado.

Definición de estrategia de validación:

La separación del conjunto de datos se hará según la variable Año_natural, la diferencia con la anterior entrega es que en este caso en vez de una estrategia de validación con k-folds, haremos entrenamiento con los años 2019 y 2020, validación con el año 2021, y test con 2022, lo que nos dará una visión más realista del funcionamiento de nuestro modelo para próximas temporadas.

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.

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. Esto lo hemos hecho con StandardScaler().

3. Análisis de variables relevantes

Basándonos tanto en la fase de exploración en la correlación con la respuesta, 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 a priori deberían ser: *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 47 de variables pega una bajada significativa de aproximadamente 200k, pasa de aproximadamente 3,135,194.71, a 2,922,583.36, aunque esta técnica en comparación con kBest no funciona nada bien, por lo que continuaremos con esta última.

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 20 variables, incluyendo: *'Mins', 'Goals', 'Assists', 'SpG', 'PS%','MotM', 'KeyP', 'OffDrb', 'AvgP', 'ThrB', 'Titularidades', 'Equipo_pos','1_año_anterior','2_año_anterior','3_año_anterior','4_año_anterior','5_año_anterior','Equipo_FC Barcelona', 'Equipo_Real Madrid', 'nationality_Germany',* 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, con este número de variables da un error de 2,653,829.52, esta propuesta es interesante, porque aparte de coger las variables típicas, tiene en cuenta la nacionalidad alemana, lo cual resulta curioso.

Adicionalmente, un apunte que queremos mencionar, es que, con 17 variables, aparte de ser el segundo mejor punto y coger variables parecidas a las que coge con 20, 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 un MAE de 2,661,513.86.

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

Ajuste de Hiperparámetros:

No tiene sentido hacer ajuste de hiperparámetros, porque es una regresión lineal y no tiene hiperparámetros que ajustar.

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 20 variables seleccionadas en la fase anterior, mientras que en el segundo caso se utilizaron todas. Los resultados son los siguientes, el modelo con las 20 variables obtuvo un MAE de 2,653,829.52 en el conjunto de validación, mientras que el modelo con todas las variables dio un MAE de 2,865,652.99. 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, añadir también que esta separación entre años nos ha beneficiado, pues el mejor error antes era 2,787,930.65.

Hemos realizado el escalado de los datos, y ha resultado, en que en el modelo final escalado da 2,564,070.29 y sin escalar 2,567,496.50, ambos en conjunto de test

Se realizó un análisis por segmentos separando los datos por posición para la regresión lineal, y resultó en lo siguiente, el modelo final completo tiene las siguientes predicciones, para defensas 2,263,098.17, para medios 3,369,311.47, para delanteros 2,376,611.74 y para porteros 2,247,259.78, mientras que separado para cada una de las posiciones, ajustado con el mejor número de variables para cada uno da para defensas 2,387,188.35 para medios 3,127,964.94, para delanteros 2,965,614.53, para porteros 1,911,311.08.

También se hizo un análisis de error por precios, que se hizo para ver si el modelo podía ser muy bueno por ejemplo para venderlo a clubes pequeños con muy buenos resultados, pero con los resultados que obtuvimos no conviene, pues para los jugadores de 0 a 1 M da 752,134.41, de 1 a 5 M da 1,173,293.52, de 5 a 10 M da 2,141,189.58, de 10 a 20 M da 3,258,661.69, de 20 a 40 M da 6,934,176.42, y por último de 40 a 80 M da 16,864,778.96, los resultados no son satisfactorios para considerar una venta para un rango pequeño, el rango del que más podemos promocionar es de 10 a 20 M, pues es el que mejor error da en comparación con el tamaño de los intervalos.

Hicimos el análisis de las variables más relevantes con el feature importance para el modelo final de 20 variables, y por orden da la siguiente importancia: *Equipo_Real Madrid, ThrB, nationality_Germany, OffDrb, MotM, Goals, Equipo_FC Barcelona, Assists, AvgP, PS%, Mins, 1_año_anterior, 4_año_anterior, 3_año_anterior, 2_año_anterior, 5_año_anterior, Titularidades, SpG, Equipo_pos, KeyP.*

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 3,178,044.89, con RandomizedSearch da 3,210,241.15y sin parámetros da 3,664,091.68. 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", a priori estos números parecen peores que la anterior separación, pero ya veremos que al final da mejores resultados.

Evaluación del modelo:

Con respecto al comportamiento del modelo en sí, con los hiperparámetros da un MAE de 2,851,779.35, que mejora bastante en comparación con el 3174704.14 que daba anteriormente, por lo que quedamos satisfechos, aunque no de mejores resultados que el de regresión lineal, pero que puede resultar bastante factible, sabiendo que muchas de las relaciones son lineales.

Por otra parte, las variables a las que les da importancia, por orden son algunas como: *'1_año_anterior', 'Mins', '4_año_anterior', 'age', 'Fouls', '2_año_anterior', 'Crosses', 'Clear', 'Goals', '5_año_anterior', 'Equipo_pos', 'Inter', 'Titularidades', 'Tackles', 'height', 'AvgP', 'SpG', 'Fouled', 'DeffDrb', 'PS%', 'position_Centre-Back', 'OffDrb', 'UnsTchBlocks', 'LongB'*, muchas de ellas son las que esperábamos.

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 2878005.19, con RandomizedSearch da 2629609.35 y sin parámetros da 2900437.35. Los parámetros que vamos a utilizar para evaluar el modelo son: solver="adam”, alpha = 0.01, batch_size = 16, learning_rate_init = 0.1, hidden_layer_sizes = (185,).

Evaluación del modelo:

Haciendo la comparativa con datos escalados y no escalados obtenemos que con el mejor modelo escalado (obtenido con RandomSearch) obtenemos un error de 2629609.35, mientras que con el mejor modelo escalado obtenemos un error de 2714148.26, frente al modelo escalado sin parámetros que da un error de 2900437.35 todo en el conjunto de test.

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.