Análisis de los jugadores de las cinco grandes ligas europeas de fútbol mediante técnicas de minería de datos en la temporada 2019-2020

Asignatura: Minería de datos

Máster Universitario en Ingeniería de Análisis de Datos, Mejora de Procesos y Toma de Decisiones (UPV)

Autores: Javier Porcel Marí
José Nicolás Granero Moreno

Contenido

Resumen

Este trabajo consiste en analizar diferentes datos referentes a los jugadores que participaron en la temporada de fútbol 2019-2020 en las cinco grandes ligas europeas (Premier League en Inglaterra, Ligue 1 en Francia, Bundesliga en Alemania, Serie A en Italia y La Liga en España) mediante técnicas de minería de datos tanto supervisadas como no supervisadas. Para las técnicas supervisadas, se buscará clasificar a los jugadores por posiciones y con algunas de éstas, se tratará de encontrar que características son diferenciadoras para realizar esta clasificación.

1. Introducción

Hoy en día se vive una gran revolución digital en la que el análisis de datos forma parte de la toma de decisiones de la gran mayoría de organizaciones volviendo a estas más competitivas. Entre este tipo de organizaciones se encuentran los clubs de fútbol. Concretamente, el análisis de datos pasa a ser una buena herramienta para formar plantillas de jugadores bien balanceadas como para aprovechar buenas oportunidades que se ofrecen en el mercado de fichajes.

De estas nuevas necesidades del fútbol actual, surge la motivación de este trabajo en el que se busca realizar un análisis profundo mediante técnicas de minería de datos, tanto mediante técnicas de aprendizaje no supervisado como mediante técnicas de aprendizaje supervisado, de las características que poseen los diferentes jugadores que participaron en la temporada futbolística de las cinco grandes ligas europeas en los años 2019-2020. Estas cinco ligas de fútbol analizadas son: Premier League (Inglaterra), Ligue 1 (Francia), Bundesliga (Alemania), Serie A (Italia) y La Liga (España).

Figura 1. Logos de las grandes ligas europeas.

Concretamente con las técnicas no supervisadas (Fuzzy clustering), se buscará realizar un análisis exploratorio de los datos. Y con las técnicas supervisadas, se tratará de clasificar a los jugadores por posición. Además de las técnicas supervisadas vistas en la asignatura de “Minería de Datos” (tales como support vector machine, árbol de clasificación, random forest, vecinos más próximos, Naive Bayes, Bagging y Boosting), se utilizarán también técnicas vistas en las asignaturas de “Simulación y Redes Neuronales” y “Análisis, Monitorización y Diagnóstico de Procesos Multivariantes”. Estas técnicas adicionales engloban redes neuronales, SIMCA y PLS-DA. Todas las técnicas mencionadas menos SIMCA y PLS-DA toman como datos los scores de un PCA previo. Además es importante mencionar que los pocos datos faltantes que hay en la base de datos se han imputado mediante la técnica knn.

2. Metodología

2.1. Fuzzy Clustering (aprendizaje no supervisado)

La única técnica de aprendizaje no supervisado que se ha utilizado en este trabajo es la técnica Fuzzy Clustering que consiste en dividir los individuos en varios clústeres o grupos pudiendo un individuo pertenecer a más de un clúster (partición suave restringida), a diferencia de lo que sucede con el clustering clásico donde un individuo solo puede pertenecer a un único clúster (partición dura) .

Como ya se ha mencionado anteriormente, se realiza un PCA previo de los datos y nos quedamos con 10 componentes principales. De este PCA se cogen los scores para realizar con ellos el Fuzzy Clustering. Para realizar el PCA (que se aplicará no solo para el Fuzzy Clustering) se ha utilizado el paquete “stats” (la función princomp) del software libre R y para realizar los clústeres Fuzzy se ha utilizado el paquete “cluster” del mismo software (la función fanny).

2.2. Técnicas supervisadas de clasificación

2.2.1. Support Vector Machine

Una máquina de soporte vectorial es un algoritmo de aprendizaje supervisado que sirve para clasificar datos mediante la búsqueda de un hiperplano que separa de forma óptima los puntos de una clase de la de otra. Esta ilustración nos da una primera intuición de los comentado anteriormente:

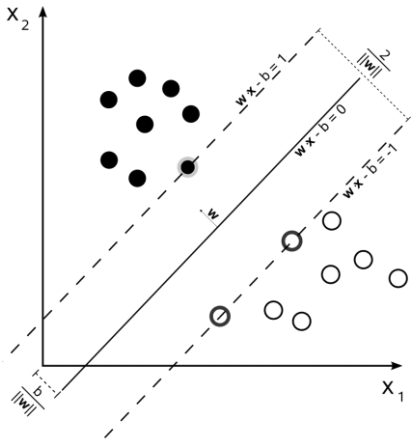

Figura 2. Ilustración del funcionamiento del algoritmo SVM.

Para implementar esta técnica de clasificación, se han utilizado los paquetes “kernlab” (la función ksvm) y “e1071” (la función svm) del software estadístico libre R.

2.2.2. Naive Bayes

El algoritmo de clasificación Naive Bayes se basa en el teorema de Bayes para cumplir su función. El teorema de Bayes tiene la siguiente expresión:

$$P(C/X_1, X_2, \ldots, X_p) = \frac{P(X_1, X_2, \ldots, X_p/C)P(C)}{P(X_1, X_2, \ldots, X_p)}$$

donde C es la clase y las diferentes X son las variables predictoras.

Para implementarlo en R se ha utilizado el paquete “e1071” y de este paquete se ha hecho uso de la función naiveBayes que supone independencia y una distribución gaussiana. Además se ha implementado el algoritmo Naive Bayes con suavizado de Laplace, lo que se consigue al añadir a la función naiveBayes el parámetro laplace = 3.

2.2.3. Vecinos más próximos

La técnica de los k vecinos más cercanos se puede usar tanto para clasificación como para regresión (aunque en este trabajo solo será utilizada para clasificar) y para clasificación consiste en predecir una variable categórica al observar cuál es la clase más frecuente de los k puntos más cercanos al punto que se quiere predecir. Para determinar qué puntos son los más cercanos, debe definirse una métrica que nos permita obtener distancias. En este trabajo se ha utilizado k = 3 en la función knn del paquete “class” del software libre R. Esta función utiliza la distancia euclídea para determinar los vecinos más próximos.

2.2.4. Árbol de clasificación

El árbol de clasificación es una técnica de aprendizaje supervisado de clasificación que consiste en crear “ramificaciones” y “hojas” de manera que cada ramificación o nodo especifica el test de algún atributo concreto por lo que según el camino que siga el individuo cuya variable categórica se está tratando de predecir acabará en una “hoja” u otra lo que indicará la clase que se debe asociar a la predicción. Para esta técnica se ha utilizado la función rpartXse del paquete “DMwR” del software R. También se han utilizado los paquetes “rpart” y “rpart.plot”.

2.2.5. Bagging y boosting

Los algoritmos bagging y boosting se basan en combinar el uso de diferentes clasificadores. Para el algoritmo bagging cada clasificador se induce independientemente además de inducirse una fase de diversificación sobre los datos mientras que para el algoritmo boosting cada clasificador tiene en cuenta los fallos del anterior. Para implementar ambos algoritmos se ha hecho uso de las funciones bagging y boosting del paquete “adabag”.

2.2.6. Random Forest

El random forest o bosque aleatorio consiste en la creación de árboles de clasificación en los que los nodos no se producen a partir del conjunto total de predictores, sino que se producen mediante un subconjunto elegido al azar. Para la utilización de esta técnica, se ha hecho uso de la función randomForest del paquete “randomForest” del software R.

2.2.7. Otras técnicas

Además de las técnicas propias de la asignatura, se han implementado otras tres técnicas de clasificación.

En primer lugar, una red neuronal: algoritmo que está formado por un conjunto de unidades llamadas neuronas (en nuestro caso se habrán implementado 60 neuronas en una única capa oculta con un threshold de 0,01) que están conectadas entre sí y acaban proporcionando unos valores de salida (función neuralnet del paquete “neuralnet”).

Por otro lado, se utilizará un PLS discriminante (función plsda del paquete “mdatools”) que consiste en obtener componentes que maximicen la covarianza entre las variables predictoras y las variables a predecir (llamadas componentes PLS) y realizar una regresión sobre estas componentes. También, se utilizará un SIMCA (función simca del paquete “mdatools”) que consiste en realizar un PCA entrenado con los datos de una clase, la validación de éste se realizará con obteniendo los “scores” de los datos de validación con los “loadings” del PCA previo y clasificando al individuo de esa clase si está por debajo de los límites para el gráfico de residuos y de la T^2 de Hotelling. En caso contrario, el individuo será una observación atípica o extrema y se clasificará como que no pertenece a esa clase.

2.3. Validación

2.3.1. Hold out y hold out repetido

La primera de las técnicas implementadas para validar las técnicas de clasificación explicadas anteriormente es la técnica hold out que consiste en dividir aleatoriamente el conjunto de datos en dos subconjuntos: el subconjunto de entrenamiento y el subconjunto de validación (en este trabajo se han partido los datos para que el 75% sirvan para entrenar los modelos y el 25% para validarlos). Una vez divididos los datos, se procede a entrenar los modelos y posteriormente a observar sus tasas de acierto.

Una modificación del hold out que se ha implementado es el hold out repetido que consiste en repetir el hold out convencional varias veces cambiando la partición de los datos de manera que la tasa de acierto total será el promedio de la tasa de acierto para cada iteración. Concretamente se han realizado 100 iteraciones.

2.3.2. Validación cruzada

La validación cruzada consiste en dividir los datos en k subconjuntos de igual tamaño (en nuestro caso k = 10) y se utilizará para cada iteración uno de los subconjuntos para validar y el resto para entrenar los modelos. De este modo, se consigue que los conjuntos de validación no se solapen.

3. Resultados

3.1. Descripción de la base de datos

La base de datos que se analiza en este trabajo está formada por un conjunto de datos relativos a los jugadores de fútbol que juegan en las principales ligas de fútbol europeas en la temporada 2019-2020. Toda la información se ha obtenido de la siguiente página web: www.fbref.com (página web en la que se documentan estadísticas, resultados e historia de más de 100 competiciones de equipos tanto masculinos como femeninos).

Así pues las variables que proporciona esta base de datos, junto con una explicación de la misma, son:

- **RL**: número de identificación del jugador.
- **Jugador**: nombre y primer apellido del jugador.
- **Pais**: nacionalidad del jugador.
- **Posc**: posición en la que juega el jugador (pudiendo ser PO para portero, DF para defensa, CC para centrocampista y DL para delantero).
- Equipo: club de fútbol en el que juega el jugador.
- **Comp**: liga en la que juega el jugador.
- **Edad**: edad del jugador en años.
- Nacimiento: año de nacimiento del jugador.
- **PJ**: número de partidos jugados por el jugador.
- **Titular**: número de partidos jugados desde el inicio por el jugador.
- **Min**: minutos jugados por el jugador.
- **90s**: minutos jugados por el jugador divididos entre 90 ya que 90 minutos equivale a un partido en términos de duración.
- **G_TP**: goles que ha marcado el jugador en toda la temporada por cada 90 minutos jugados descontando los penaltis.
- **Dis**: disparos que realiza el jugador por cada 90 minutos jugados descontando los penaltis.
- **DaP**: disparos a puerta que realiza el jugador por cada 90 minutos jugados descontando los penaltis.
- **Dist**: distancia promedio a la que el jugador dispara medida en yardas.
- **npxG**: goles esperados que no sean de penalti por cada 90 minutos jugados.
- **np:G_xG**: goles marcados por el jugador por cada 90 minutos jugados que no sean de penalti menos los goles esperados por el jugador por cada 90 minutos que no sean de penalti.
- **Pases_Comp.**: pases completados por el jugador por cada 90 minutos jugados.
- **Pases_Int.**: pases intentados por el jugador por cada 90 minutos jugados.
- **Dist.tot.:** distancia promedio, en yardas, que han recorrido los pases completados en cualquier dirección por cada 90 minutos jugados.
- **Dist._prg.:** distancia promedio, en yardas, que han recorrido los pases completados hacia la meta del oponente por cada 90 minutos jugados.

- **Cortos_Cmp.:** pases completados a una distancia de entre 5 y 15 yardas por el jugador por cada 90 minutos jugados.
- **Cortos_Int..:** pases intentados a una distancia de entre 5 y 15 yardas por el jugador por cada 90 minutos jugados.
- **Medios_Cmp.:** pases completados a una distancia de entre 15 y 30 yardas por el jugador por cada 90 minutos jugados.
- **Medios_Int.:** pases intentados a una distancia de entre 15 y 30 yardas por el jugador por cada 90 minutos jugados.
- **Largos_Cmp.:** pases completados a una distancia de más de 30 yardas por el jugador por cada 90 minutos jugados.
- **Largos_Int.:** pases intentados a una distancia de de más de 30 yardas por el jugador por cada 90 minutos jugados.
- **Ass**: asistencias realizadas por el jugador por cada 90 minutos jugados.
- **xA**: asistencias esperadas por el jugador por cada 90 minutos jugados.
- **A_xA**: asistencias realizadas por el jugador por cada 90 minutos jugados menos las asistencias esperadas en ese lapso de tiempo.
- **Tkl**: número de entradas realizadas por el jugador por cada 90 minutos jugados.
- **TklG**: número de entradas ganadas por el jugador por cada 90 minutos jugados.
- **Tkl_def**: : número de entradas realizadas por el jugador por cada 90 minutos jugados en la zona defensiva (1/3 del campo más cercano a la portería del equipo).
- **Tkl_cent**: número de entradas realizadas por el jugador por cada 90 minutos jugados en el mediocampo (1/3 del campo más cercano al centro del campo).
- **Tkl_ataq**: número de entradas realizadas por el jugador por cada 90 minutos jugados en la zona ofensiva (1/3 del campo más cercano a la portería rival).
- **Presion**: número de veces que se aplica presión al jugador del equipo rival que está recibiendo, llevando o soltando la pelota por cada 90 minutos jugados.
- **Pres_exito**: número de veces que se aplica presión al jugador del equipo rival que está recibiendo, llevando o soltando la pelota por cada 90 minutos jugados y se consigue recuperar el balón tras cinco segundos de presión.
- **Pres_def**: número de veces que se aplica presión al jugador del equipo rival que está recibiendo, llevando o soltando la pelota por cada 90 minutos jugados en la zona defensiva (1/3 del campo más cercano a la portería del equipo).
- **Pres_cent**: número de veces que se aplica presión al jugador del equipo rival que está recibiendo, llevando o soltando la pelota por cada 90 minutos jugados en el mediocampo (1/3 del campo más cercano al centro del campo).
- **Pres_ataq**: número de veces que se aplica presión al jugador del equipo rival que está recibiendo, llevando o soltando la pelota por cada 90 minutos jugados en la zona ofensiva (1/3 del campo más cercano a la portería rival).
- **Int**: número de intercepciones por cada 90 minutos jugados.

- **Tkl+Int**: número de entradas más intercepciones por cada 90 minutos jugados.
- **Desp.:** número de despejes por cada 90 minutos jugados.
- **TA**: número de veces que el jugador es penalizado con una tarjeta amarilla por cada 90 minutos jugados.
- **TR**: número de veces que el jugador es penalizado con una tarjeta roja por cada 90 minutos jugados.
- **AereoG**: número de duelos aéreos ganados por cada 90 minutos jugados.
- **AereoP**: número de duelos aéreos ganados por cada 90 minutos jugados.

Las variables “Edad” y “Nacimiento” aportan la misma información a la base de datos por lo que solo se utilizará la variable “Edad” en las diferentes técnicas de aprendizaje automático de este trabajo.

A pesar de que la base de datos original posee cuatro posiciones (porteros PO, defensas DF, centrocampistas CC y delanteros DL) se trabajará eliminando a los porteros por lo que se trabajará solo con defensas, centrocampistas y delanteros. Esto se debe a que debido a la función totalmente diferenciadora que realizan los porteros las variables que los definen son completamente distintas a las que definen al resto de los jugadores.

Además se ha decidido eliminar del análisis a todos aquellos jugadores que no haya jugado durante la temporada un mínimo de 900 minutos lo que equivaldría a 10 partidos de fútbol jugados ya que se considera que los datos que pertenecen a esos jugadores no son lo suficiente robustos.

Para matizar las variables Tkl_def, Tkl_cent, Tkl_ataq, Pres_def, Pres_cent y Pres_ataq se muestran mediante un diagrama las tres zonas a las que se refieren estas seis variables (es decir, se ha divido el campo en tres rectángulos iguales de modo que Tkl_def y Pres_def se han medido en el rectángulo más cercano a la portería que hay que defender; Tkl_cent y Pres_cent se han medido en el rectángulo correspondiente al centro del campo; y Tkl_ataq y Pres_ataq se han medidoen el rectángulo más cercano a la portería rival).

Cabe mencionar para completar la explicación del siguiente diagrama que suponiendo que el jugador analizado perteneciera al equipo rojo, las tres zonas que aparecen en el dibujo serían de abajo a arriba, la zona defensiva, la zona del medio campo y la zona ofensiva.

Figura 3. Dibujo representativo de la división del campo de juego en tres zonas.

Por último es necesario explicar a qué se refieren las variables npxG y xA cuando se refieren a goles y asistencias esperadas, aunque estos valores no se calculen en este trabajo sino que se obtienen directamente de la fuente. Estos dos valores se obtienen mediante la suposición de que todas las ocasiones de gol poseen una cierta probabilidad de gol para todos los jugadores que realicen el disparo. Así pues está probabilidad dependerá de la distancia a la portería, el ángulo respecto a la portería, parte del cuerpo con la que se realiza el remate, tipo de asistencia (pase corto, pase en profundidad, pase al hueco, centro, regate, balón parado), etc. De este modo se asocia unos goles y unas asistencias esperadas para cada jugador según las oportunidades que haya tenido. Aquel jugador que sea capaz de batir los valores esperados con sus valores reales será considerado un jugador eficaz pues aprovecha las oportunidades que se le presentan.

A modo ilustrativo y como ejemplo se muestra un diagrama donde se observan las diferentes probabilidades de anotar un tanto cuando un jugador realiza un remate de cabeza a partir de un centro al área:

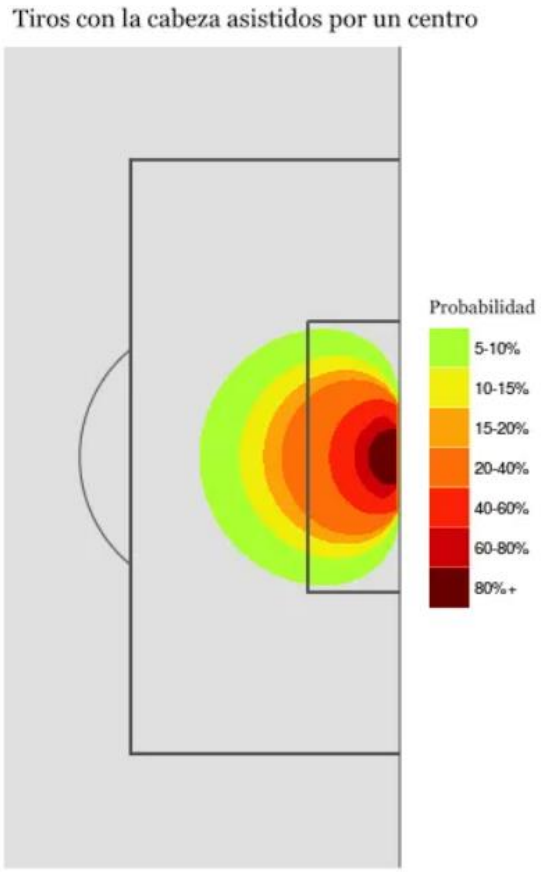

Figura 4. Ilustración de la probabilidad de gol para un remate de cabeza a partir de un centro al área según la zona de remate.

Una vez aclarada toda la información que se encuentra en la base de datos se explican en los siguientes apartados las diferentes técnicas de minería de datos que se han utilizado en este trabajo y se muestran los resultados obtenidos en cada una de ellas.

3.2. Fuzzy Clustering (aprendizaje no supervisado)

Se han realizado diferentes pruebas con Fuzzy clustering variando la cantidad de clústeres y variando las distancias (se ha probado con la distancia euclídea, con la distancia de manhattan y con la distancia euclídea al cuadrado) obteniéndose en todos los casos clústeres muy solapados por lo que los individuos poseen unos coeficientes de pertenencia a los diferentes clústeres excesivamente similar. De esto se concluye que esta herramienta no nos ha permitido obtener información y no nos ha sido útil.

A modo ilustrativo de los dicho anteriormente, se muestra la gráfica para cuando se busca obtener tres clústeres mediante la distancia euclídea al cuadrado (concretamente esta configuración corresponde a la función fanny, del paquete cluster, con los parámetros k = 3 y metric = “SqEuclidean”) :

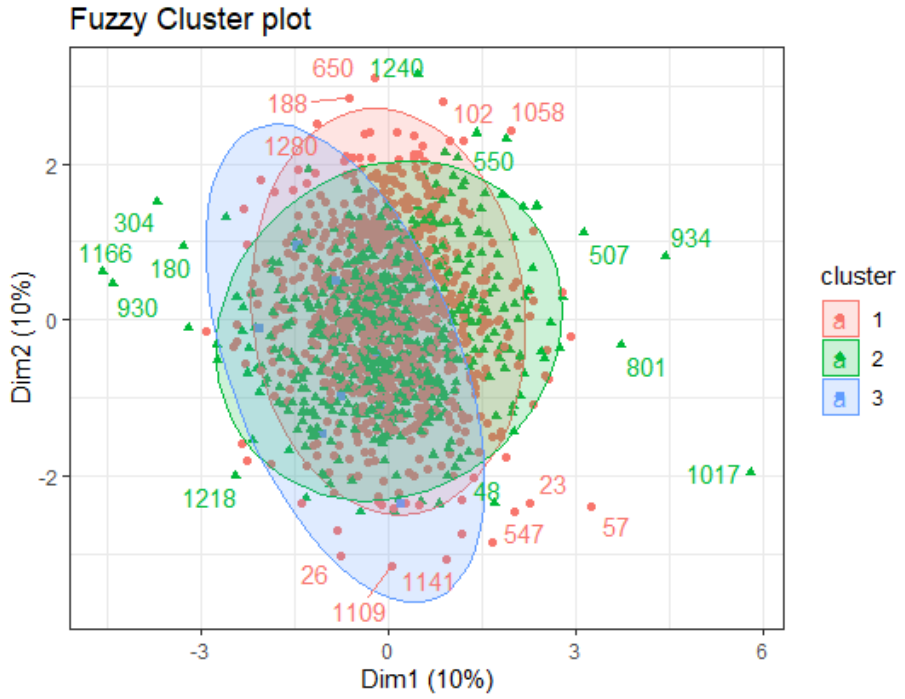

Figura 4. Gráfica con los tres clústeres Fuzzy.

3.3. Técnicas supervisadas de clasificación

3.3.1. Hold out

En este método, se ha realizado cogiendo una muestra de los datos. Como se ha dicho previamente, se ha elegido una muestra cercana al 75%, al ser el número de datos de 1437 se han cogido 1079, para entrenar los modelos y 358 para validarlos.

Dentro de éstas técnicas de aprendizaje supervisado, existen algunas que indican las variables que tienen una mayor capacidad de clasificación como el Árbol de Clasificación y el Random Forest (Bosque Aleatorio), que como se puede ver en las siguientes imágenes, estas componentes son la primera y la segunda del PCA para ambos métodos:

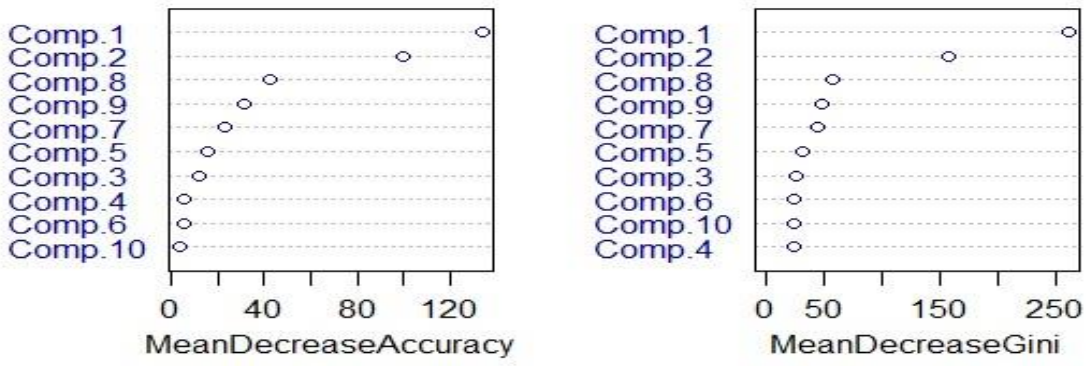

Figura 5. Gráficas que muestran la importancia de las variables del Random Forest.

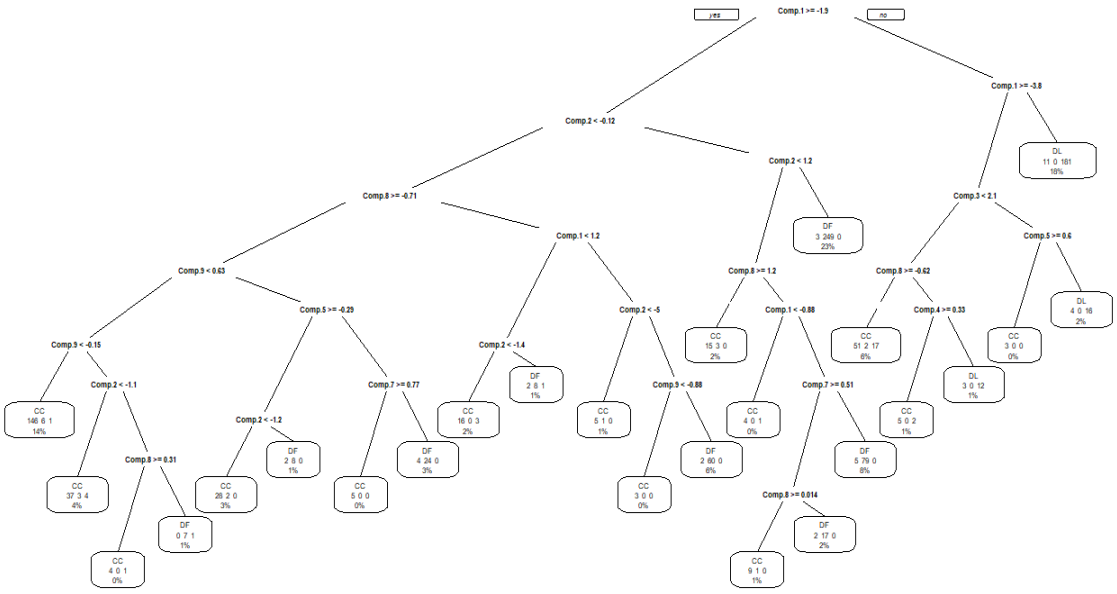

Figura 6. Árbol de clasificación.

Las tasas de aciertos de los modelos anteriores son las siguientes:

Método	Árbol	knn	n.bayes	bayes.laplace	Svm1	Svm2
Tasa aciertos	0.821	0.8407	0.855	0.855	0.866	0.883

Método	RF	NN	pls	simca	bag	boost
Tasa aciertos	0.883	0.8743	0.8685	0.438	0.838	0.877

Tabla 1. Tasas de aciertos para cada técnica después del hold out.

3.3.2. Validación Cruzada

En esta técnica de validación, se han realizado 10 carpetas de un tamaño similar, 9 de tamaño 143 y una de tamaño 150, siendo éstas carpetas totalmente diferentes entre sí y realizando el entrenamiento del algoritmo con 9 de las 10 carpetas y validándolo con la carpeta restante. De esta manera, todas las observaciones forman parte del conjunto de validación y del conjunto de entrenamiento.

A continuación, se ha mirado las tasa de acierto en media de los métodos de clasificación, un intervalo de confianza al 95% para cada método y un boxplot.

método	tree	knn	NaiveBayes	N.B.Laplace	SVM1	SVM2
tasa media	0.83511	0.85243	0.86975	0.86975	0.87328	0.89629

método	RF	NN	PLS	Simca	Bagging	Boosting
tasa media	0.8796	0.87055	0.8638	0.43198	0.86222	0.8873

Tabla 2. Tasas de aciertos para cada técnica después de la validación cruzada.

El intervalo de confianza al 95% para los métodos es el siguiente:

método	tree	knn	NaiveBayes	N.B.Laplace	SVM1	SVM2
inferior	0.79493	0.78724	0.8314	0.8314	0.82989	0.86713
superior	0.87412	0.91136	0.9001	0.9001	0.90594	0.9285

método	RF	NN	PLS	Simca	Bagging	Boosting
inferior	0.8547	0.8407	0.8393	0.4165	0.83217	0.8685
superior	0.9113	0.8989	0.9064	0.4521	0.89895	0.9090

Tabla 3. Intervalos de confianza al 95% para cada técnica después de la validación cruzada.

En este gráfico, se muestra un boxplot de las tasas de acierto de cada método:

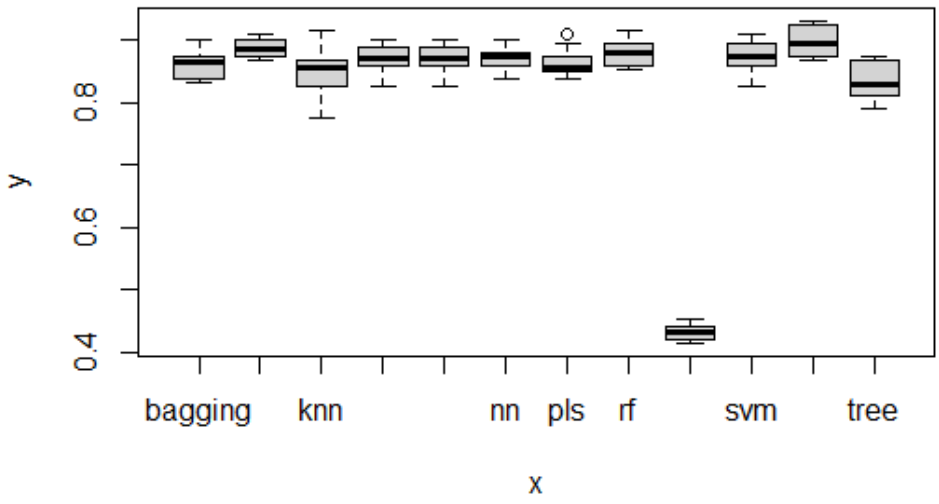

Figura 7. Boxplot de las tasas de acierto para cada técnica tras la validación cruzada (técnicas de izquierda a derecha: bagging, boosting, knn, N.Bayes, N.Bayes Laplace, nn, pls-da, Random Forest, SIMCA, SVM1, SVM2 y árbol de clasificación).

3.3.3. Hold out repetido

En este caso, se ha realizado 100 veces un hold out con unos tamaños de muestra iguales a los del hold out, citado previamente. Al igual que en la validación cruzada, se ha obtenido la tasa de acierto media, los intervalos de confianza y un boxplot.

Método	Tasa de acierto
Árbol de Clasificación	0.83569832
KNN	0.85432961
Naive Bayes	0.87075418
Naive Bayes Laplace	0.87075418
Support Vector Machine 1	0.86547486
Support Vector Machine 2	0.88918994
Random Forest	0.88075418
Neural Network	0.86460893
PLS	0.83311227
Simca	0.43133841
Bagging	0.84983240
Boosting	0.88209497

Tabla 4. Tasas de aciertos para cada técnica después del hold out repetido.

El intervalo de confianza es:

Método	Límite Inferior	Límite Superior
Árbol de Clasificación	0.80167598	0.86459497
KNN	0.82108939	0.89120111
Naive Bayes	0.84357542	0.89532123
Naive Bayes Laplace	0.84357542	0.89532123
Support Vector Machine 1	0.83519553	0.89252793
Support Vector Machine 2	0.85195531	0.91766760
Random Forest	0.85195531	0.90502793
Neural Network	0.83093575	0.89252793
PLS	0.72535071	0.89875998
Simca	0.41445122	0.45123250
Bagging	0.81696927	0.88414804
Boosting	0.84916201	0.90928771

Tabla 5. Intervalos de confianza al 95% para cada técnica después del hold out repetido.

El Boxplot es el siguiente:

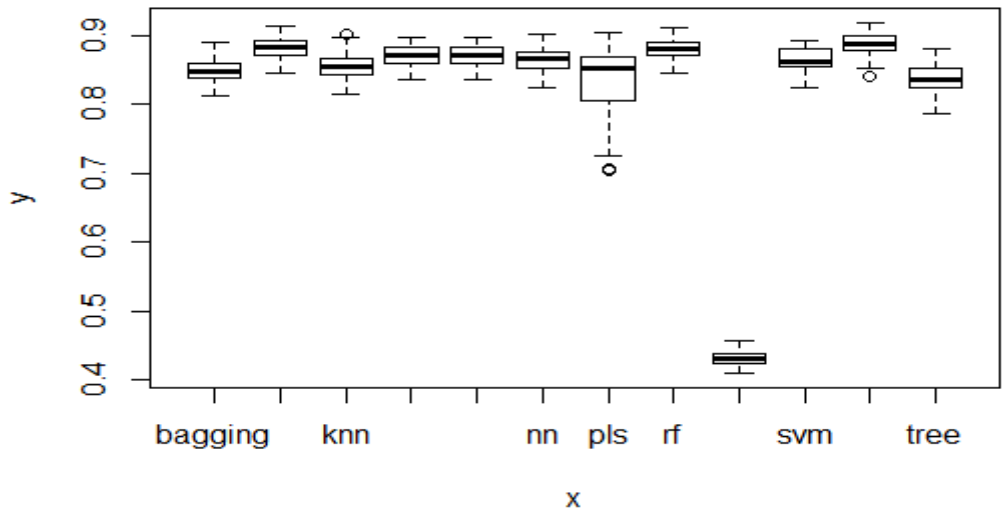

Figura 8. Boxplot de las tasas de acierto para cada técnica tras el hold out repetido (técnicas de izquierda a derecha: bagging, boosting, knn, N.Bayes, N.Bayes Laplace, nn, pls-da, Random Forest, SIMCA, SVM1, SVM2 y árbol de clasificación).

4. Conclusión

El primer objetivo del trabajo era realizar un análisis exploratorio mediante la técnica Fuzzy Clustering pero no se han obtenido resultados concluyentes mediante esté método ya que los clústeres se solapan mucho y ya que los niveles de pertenencia a cada clúster de los diferentes jugadores son extremadamente similares.

Por otro lado y por lo que respecta al uso de técnicas supervisadas para clasificar a los jugadores por posición, se puede apreciar en los gráficos y en las tasas de aciertos de la validación cruzada que el método del boosting y el random forest tienen una tasa de acierto muy parecida y que realizando un test de la t de Student para las tasas de aciertos no obtenemos diferencias significativas. En cambio, para la red neuronal se ve como tiene un peor desempeño que el boosting ya que el test de la t da diferencias significativas como ejemplo, al igual que pasa con el support vector machine 2 (el realizado con la función ksvm) que tiene una tasa significativamente más alta que el boosting. Esto mismo se puede apreciar cuando se hace el hold out repetido, el SVM tiene una diferencia significativa respecto al boosting. Sin embargo, el SVM es un algoritmo que no se sabe las variables más importantes cuales son, con lo cual aunque para clasificar sea el mejor de todos, aunque para saber cuáles son las componentes

o variables más importantes se debería elegir el random forest ya que tiene una tasa de acierto mayor que el árbol de clasificación o el PLS-DA siendo éstas técnicas las que nos dan las importancias de las componentes o variables.

5. Bibliografía

- R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. https://CRAN.R-project.org/package=readxl
- R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- Jacob Kaplan (2020). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categorical Variables. R package version 1.6.3. https://CRAN.R-project.org/package=fastDummies
- Terry Therneau and Beth Atkinson (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15. https://CRAN.R-project.org/package=rpart
- Torgo, L. (2016). Data Mining with R, learning with case studies, 2nd editionChapman and Hall/CRC. URL: http://ltorgo.github.io/DMwR2
- Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0
- David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2020). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-4. https://CRAN.R-project.org/package=e1071
- Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis (2004). kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software 11(9), 1-20. URL: http://www.jstatsoft.org/v11/i09/
- A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22. Andrea Peters and Torsten Hothorn (2019). ipred: Improved Predictors. R package version 0.9-9. https://CRAN.R-project.org/package=ipred
- Stefan Fritsch, Frauke Guenther and Marvin N. Wright (2019). neuralnet: Training of Neural Networks. R package version 1.44.2. https://CRAN.R-project.org/package=neuralnet

- Alfaro, E., Gamez, M. Garcia, N.(2013). adabag: An R Package for Classification with Boosting and Bagging. Journal of Statistical Software, 54(2), 1-35. URL http://www.jstatsoft.org/v54/i02/.
- Kucheryavskiy S (2020). “mdatools - R package for chemometrics.” Chemometrics and Intelligent Laboratory Systems, 198. URL: https://doi.org/10.1016/j.chemolab.2020.103937.
- Sebastien Le, Julie Josse, Francois Husson (2008). FactoMineR: An R Package for Multivariate Analysis. Journal of Statistical Software, 25(1), 1-18. 10.18637/jss.v025.i01
- Alboukadel Kassambara and Fabian Mundt (2020). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7. https://CRAN.R-project.org/package=factoextra
- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2019). cluster: Cluster Analysis Basics and Extensions. R package version 2.1.0.
- [Estadísticas de la temporada 2019-2020 para las 5 grandes ligas europeas]. (s.f.). FBREF.
  URL: https://fbref.com/es/comps/Big5/2019-2020/stats/jugadores/Estadisticas-2019-2020-Las-5-grandes-ligas-europeas
  URL: https://fbref.com/es/comps/Big5/2019-2020/shooting/jugadores/Estadisticas-2019-2020-Las-5-grandes-ligas-europeas
  URL: https://fbref.com/es/comps/Big5/2019-2020/passing/jugadores/Estadisticas-2019-2020-Las-5-grandes-ligas-europeas

  URL: https://fbref.com/es/comps/Big5/2019-2020/defense/jugadores/Estadisticas-2019-2020-Las-5-grandes-ligas-europeas

  URL: https://fbref.com/es/comps/Big5/2019-2020/misc/jugadores/Estadisticas-2019-2020-Las-5-grandes-ligas-europeas

-