PROPUESTA PROYECTO FINAL MACHINE LEARNING

MACHINE LEARNING EN LA LIGA

Predicción de resultados y goles de partidos de fútbol de La Primera División de España

---

Ignacio García Santamaría

María Oliva Calero

Propuesta de Proyecto Final
Machine Learning
14/11/2021

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Tema y justificación

El fútbol es un deporte donde siempre se ha dicho que dos más dos nunca suman cuatro, es decir, que en este deporte la ciencia no tiene cabida. "Fútbol es fútbol" y algunos dicen que es imposible de predecir. Esta aleatoriedad a la hora de predecir los resultados puede justificar el hecho de que las casas de apuestas subsistan. No obstante, nosotros queremos demostrar que el fútbol sí se puede predecir, demostrando que con el uso de datos y diferentes algoritmos podemos explicar lo que va a ocurrir en el deporte rey en España.

Por ello el objetivo principal de este proyecto es explorar el problema de predicción de resultados de fútbol mediante técnicas de Machine Learning. En nuestro proyecto utilizaremos datos de los partidos jugados en las últimas 10 temporadas de La Liga. Nuestra intención es comparar los diferentes tipos de aprendizaje y algoritmos estudiados, para demostrar que en el fútbol hay factores que pueden tener más peso que otros a la hora de determinar qué equipo se alzará con la victoria.

Objetivo y necesidades que se cubrirían

Dos objetivos de análisis.

En primer lugar, se emplearán técnicas de clasificación para predecir el resultado final de un partido (Gana local, Empate o Gana Visitante). (María)

En segundo lugar, se emplearán técnicas de regresión para realizar dos predicciones, la primera es el número de goles que marcará el equipo local (FTHG) y la segunda es determinar el número de goles que marcará el equipo visitante (FTAG). (Ignacio)

Adicionalmente, de manera conjunta vamos a discutir los resultados obtenidos, analizando cual es la mejor manera de determinar el vencedor de un encuentro. Como a partir del estudio de regresión realizado por Ignacio se puede determinar el vencedor del encuentro, tenemos un punto común de análisis de los resultados obtenidos.

El análisis se hará con la herramienta RStudio y las técnicas aprendidas en clase.

A partir de este trabajo queremos aprender que factores son los más determinantes a la hora de averiguar quién será el ganador de un encuentro. De este modo, comprenderemos mejor este deporte, llegando incluso a mejorar la experiencia a la hora verlo porque nos fijaremos más en aquellos factores determinantes.

Además, si nos fijamos más en la utilidad de los modelos que van a ser obtenidos, y no tanto en la información obtenida a medida que los modelos son desarrollados. A toda persona relacionada con el mundo del fútbol le gustaría saber cuántos goles van a ser encajados o qué equipo va a ganar antes o

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

mientras se juega un partido. Asimismo, esta predicción sería basada en datos imparciales, no en la opinión condicionada de un periodista o de un analista deportivo.

Metodología de trabajo y técnicas previstas utilizar

Una vez encontrado un conjunto de datos y tras establecer los objetivos. Tanto María como Ignacio vamos a realizar nuestro estudio de manera independiente. Estableciendo cada uno nuestras propias conclusiones acerca del mejor algoritmo para predecir, cuáles son las variables más significativas o cualquier información adicional útil para el objetivo perseguido.

Una vez realizado el análisis individual, discutiremos los resultados a los que hemos llegado cada uno, poniendo en común la información adquirida y estableciendo que modelo emplearíamos en función de la situación analizada, determinando de este modo quien se ha acercado más al objetivo común de ambos estudios (determinar el desenlace de un partido).

Técnicas previstas a utilizar por parte de Ignacio: A la hora de hacer el análisis de regresión de las dos variables a predecir, voy a analizar de manera independiente cada una, pero empleando las mismas técnicas. Para hacer el análisis de la mejor manera, y lo más completo posible, voy a emplear técnicas de regresión lineal, técnicas de regresión lineal normalizadas, KNN en regresión y árboles de decisión en regresión. Tras estudiar cada método de manera particular, tengo la intención de aplicar métodos de ensemble y así poder obtener el mejor modelo a partir de las técnicas estudiadas.

Técnicas previstas a utilizar por parte de María: De manera similar a mi compañero, voy a emplear varias técnicas, para después compararlas y obtener el mejor modelo posible. Entre estas utilizaré técnicas de regresión logística, clustering, y árboles de decisión. Igual que Ignacio, después aplicaré métodos de ensemble para obtener el mejor resultado posible.

Datos

El conjunto de datos elegido recoge la información de los partidos de la Liga en las temporadas de 2010 a 2019. Con esta información, realizaremos un análisis de qué factores incluyen más en el resultado de los partidos cuantos goles serán encajados por cada equipo.

Tuvimos que depurar los datos, pues los datos de cada temporada se encontraban en archivos CSV distintos, cada uno con un número de variables diferentes. Para ello, filtramos los datos de cada temporada en base a las variables comunes, y después los unimos en un conjunto de datos único de 3800 observaciones y 34 variables.

Resultados esperados alcanzar

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Con este trabajo esperamos poder demostrar que el fútbol se puede predecir. Nuestra idea es llegar a la conclusión de que las técnicas de Machine Learning pueden ser muy útiles en este sector, no solo en la predicción de resultados y el número de goles, que es el objetivo principal de este proyecto. Sino también, en comprender qué características del juego son determinantes, de este modo, tanto los espectadores como los propios entrenadores podrán considerar determinadas variables para conseguir que su equipo gane.

Si conseguimos demostrar que mediante la aplicación de varios algoritmos podemos extraer un conocimiento que no es empleado de manera extensa por equipos de fútbol, puede que la implementación de técnicas de Machine Learning se extienda en otras áreas de este deporte, como en la elección de fichajes, por ejemplo.

Variables

Nombre	**Descripción**	**Tipo**
Season	Temporada en la que se jugó el partido	Categórica
HomeTeam	Equipo local	Categórica
AwayTeam	Equipo visitante	Categórica
Date	Fecha en la que se jugó el partido	Date
FTHG	Goles totales del equipo local en el partido	Int
FTAG	Goles totales del equipo visitante en el partido	Int
FTR	Resultado final	Categórica (H= Home win, D=Draw, A=Away Team)
HTHG	Goles del equipo local al descanso	Int
HTAG	Goles del equipo visitante al descanso	Int
HTR	Resultado al descanso	Categórica (

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

		H= Home win, D=Draw, A=Away Team)
HS	Tiros del equipo local	Int
AS	Tiros del equipo visitante	Int
HST	Tiros a puerta del equipo local	Int
AST	Tiros a puerta del equipo visitante	Int
HF	Faltas cometidas por el equipo local	Int
AF	Faltas cometidas por el equipo visitante	Int
HC	Córneres del equipo local	Int
AC	Córneres del equipo visitante	Int
HY	Tarjetas amarillas del equipo local	Int
AY	Tarjetas amarillas del equipo visitante	Int
HR	Tarjetas rojas del equipo local	Int
AR	Tarjetas rojas del equipo visitante	Int
B365H	Cuota de ganar el equipo local de la casa de apuestas Bet365	Numeric
B365A	Cuota de ganar el equipo visitante de la casa de apuestas Bet365	Numeric
B365D	Cuota de empate de la casa de apuestas Bet and Win	Numeric
BWH	Cuota de ganar el equipo local de la casa de apuestas Bet and Win	Numeric

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

BWA	Cuota de ganar el equipo visitante de la casa de apuestas Bet and Win	Numeric
BWD	Cuota de empate de la casa de apuestas Bet and Win	Numeric
WWH	Cuota de ganar el equipo local de la casa de apuestas William Hill	Numeric
WHA	Cuota de ganar el equipo visitante de la casa de apuestas William Hill	Numeric
WHD	Cuota de empate de la casa de apuestas William Hill	Numeric
VCH	Cuota de ganar el equipo local de la casa de apuestas BetVictor	Numeric
VCA	Cuota de ganar el equipo visitante de la casa de apuestas BetVictor	Numeric
VCH	Cuota de empate de la casa de apuestas BetVictor	Numeric

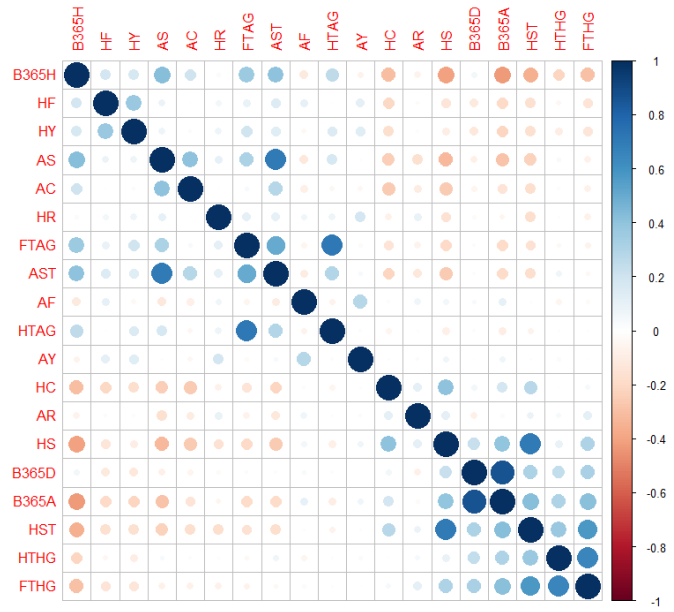
Estudio previo de los datos

En primer lugar, hacemos un análisis previo de los datos para ver qué podemos esperar y cómo se relacionan las variables.

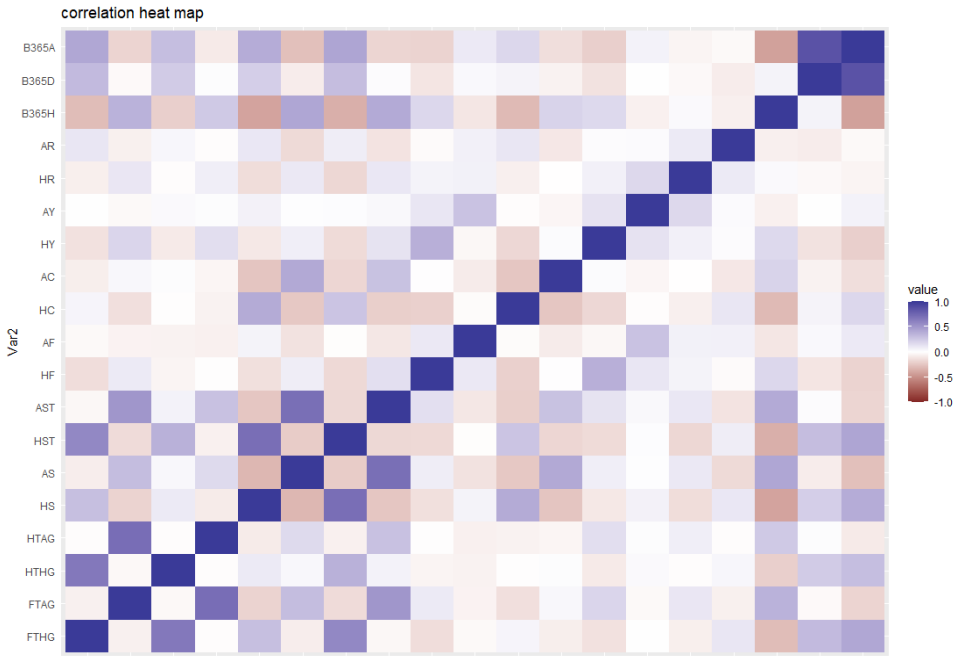
Realizando una matriz de correlación de las variables numéricas de nuestro dataframe, podemos sacar algunas conclusiones:

- Los goles totales tanto locales como visitantes están estrechamente relacionados con el número de tiros a puerta de locales y visitantes respectivamente, lo que podíamos esperar.
- Las odds de la casa de apuestas Bet365 de empate (B365D) y gana visitante (B365A) también tienen una gran relación, lo que puede llegar a sorprender.
- En el otro extremo, entre las variables que más relación negativa encontramos estaría las odds de Bet365 a que gana local (B365H) con las odds de que gane visitante, lo que encontramos lógico.
- También encontramos una relación negativa entre las odds de que gane el equipo local y el número de tiros a puerta del equipo local, un hecho sorprendente.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

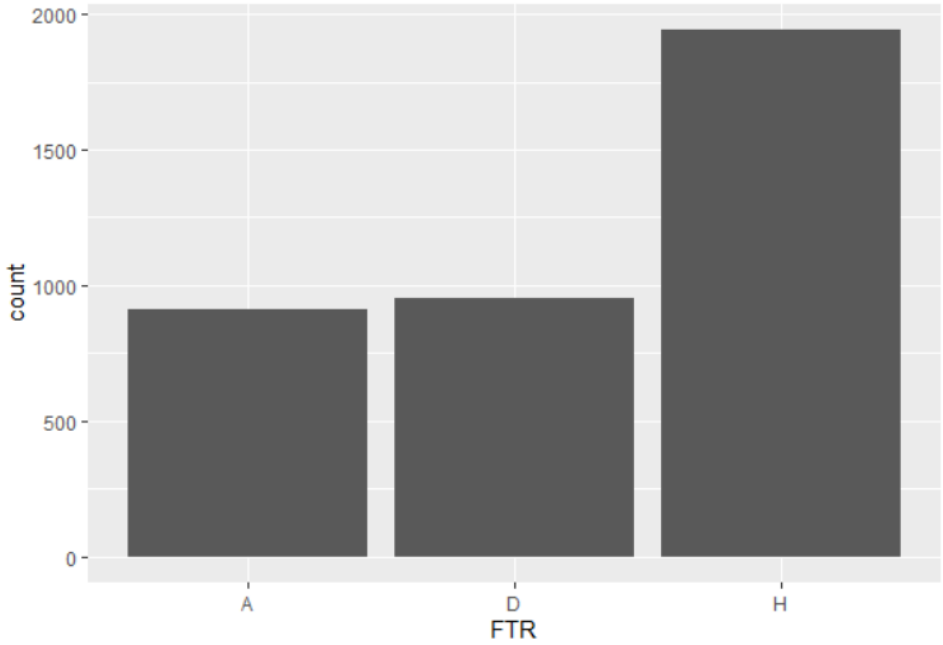


Realizamos también un heatmap que viene a corroborar el análisis de la matriz de correlación.



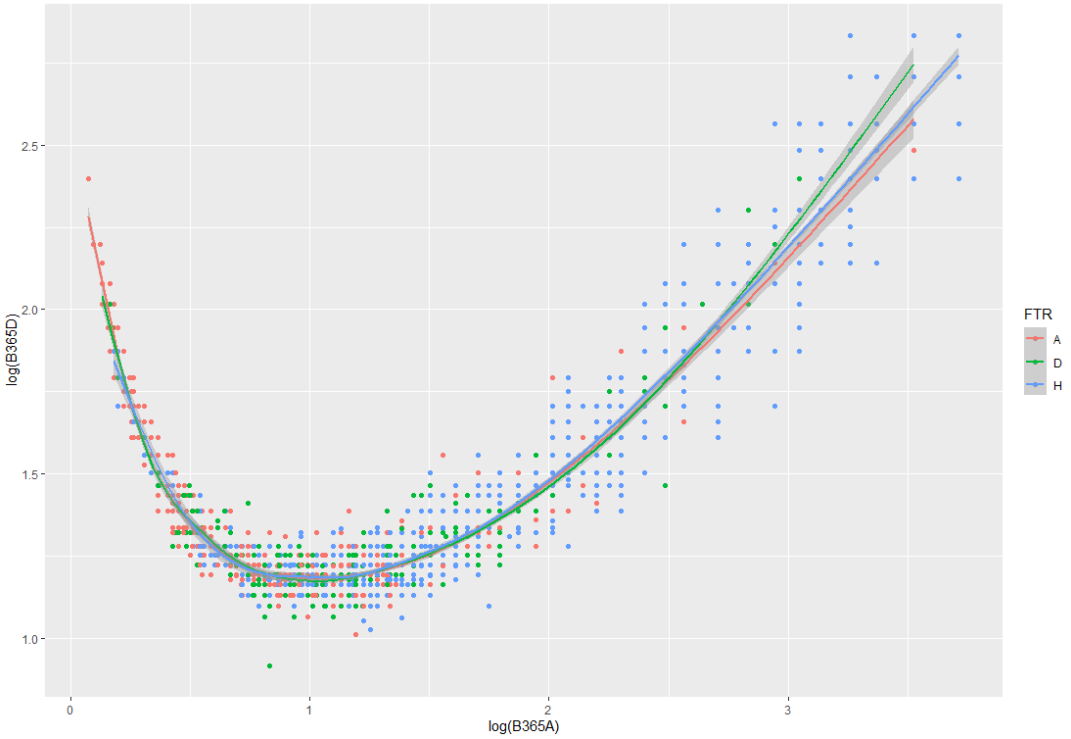
Visualizamos también el número de partidos totales que han sido ganados por equipos locales, por visitantes y que han finalizado en empate. Esto nos aporta un dato importante: la mitad de los partidos han sido ganados por equipos locales, lo que puede significar un peso importante del factor campo en el resultado de los partidos.

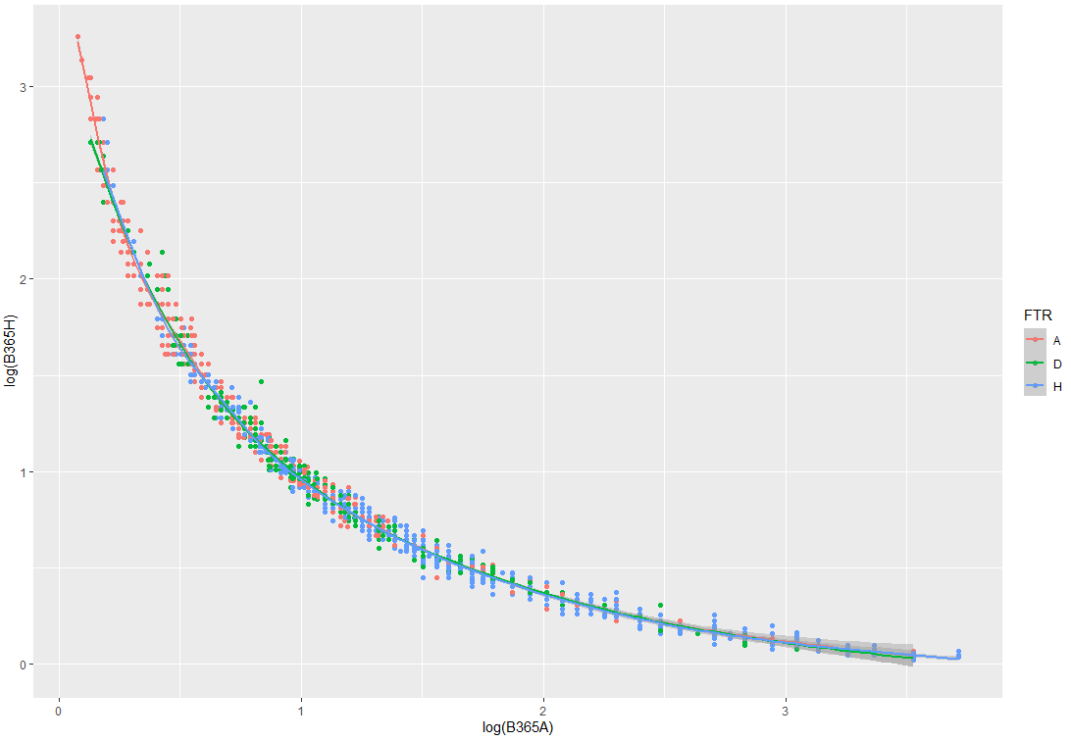
María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Exploramos también la relación entre las variables de apuestas, para demostrar que no existe correlación lineal entre ellas como hemos podido asumir por la matriz de correlación

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



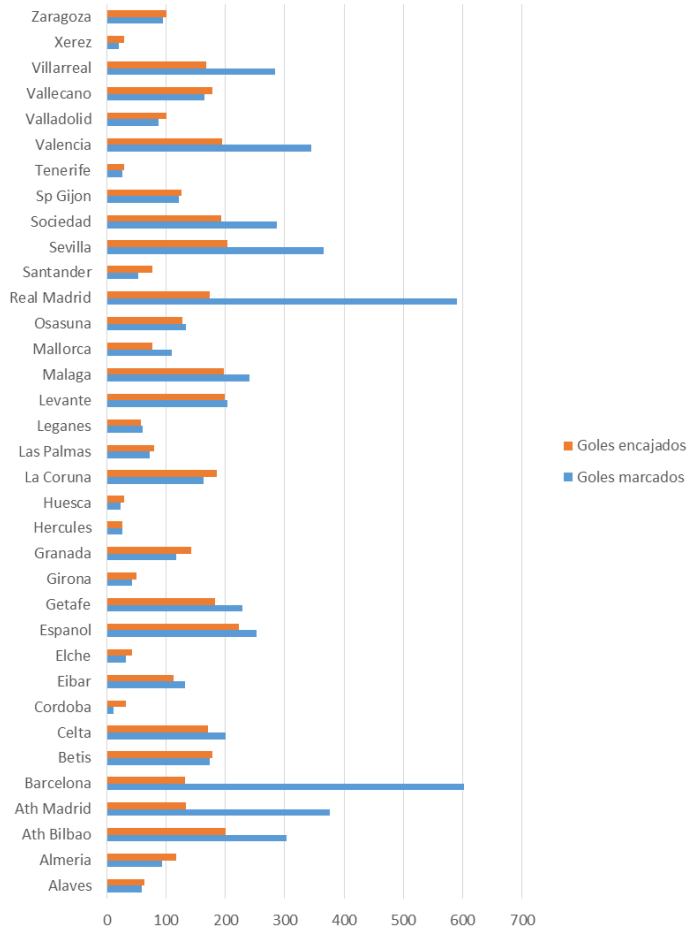


Visualizamos también la importancia que tiene jugar como local y visitante en los distintos equipos en base al número de goles anotados y encajados.

Jugando como local:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

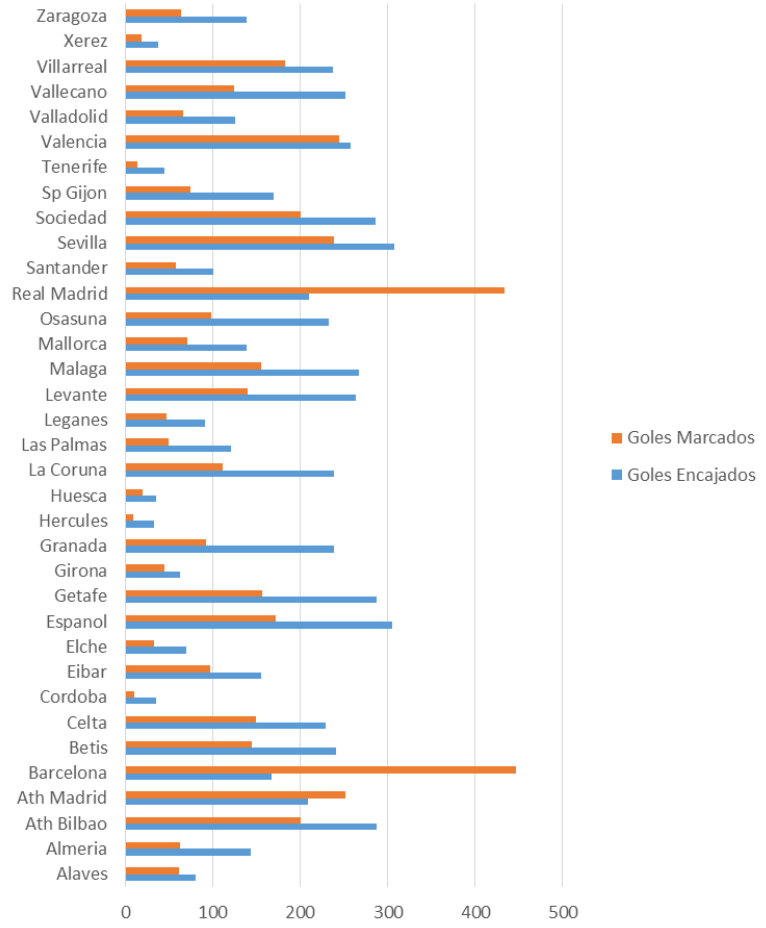
Observamos que la tendencia general es que los equipos que juegan como local marquen más goles de los encajados, con algunas excepciones.



Jugando como visitante:

En cuanto a los partidos que juegan como visitantes, podemos apreciar que los equipos suelen encajar más goles de los que marcan, a excepción de los equipos grandes como son el Real Madrid, el Barcelona, y el Atlético de Madrid. De esta información podemos llegar a la conclusión de que el equipo que juega el equipo es muy determinante a la hora de predecir el resultado del partido.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Preparación de los datos

Al pensar cómo íbamos a enfocar los datos, nos surgió un problema.

Nuestro objetivo es predecir el resultado de partidos de la liga y tenemos los recursos necesarios para modelar este problema.

Con los datos que tenemos podemos predecir el resultado de partidos que ya han sucedido, pues tenemos datos sobre los tiros a puerta, las tarjetas rojas, el número de corners... que se han dado en cada partido.

No obstante, nuestros modelos no tendrían aplicación práctica, más allá de lo académico, pues no podríamos predecir resultados de partidos futuros, pues no conocemos las estadísticas del partido antes de que se juegue.

Para resolver este problema, hemos creado, para cada partido, las medias de las diferentes variables de los últimos tres partidos jugados por ambos equipos (local y visitante).

Además, nos ha parecido interesante añadir una variable nueva (los puntos ganados por equipo en cada partido (3 en caso de victoria, 1 en caso de empate y 0 en caso de derrota), para disponer también

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

del promedio de puntos de los últimos tres partidos de los equipos que se enfrenten. Hemos creado las variables HP(Home Points) y AP(Away Points).

Para ello hemos utilizado la siguiente función:

```
get_last_3 <- function(){
  for (i in 1:nrow(df_raw)) {
    # saco los puntos de cada equipo en cada jornada
    df_raw[i,"HP" ] <- ifelse(df_raw[i,"FTR" ]=="H",3,ifelse(df_raw[i,"FTR" ]=="A",0,1))
    df_raw[i,"AP" ] <- ifelse(df_raw[i,"FTR" ]=="A",3, ifelse(df_raw[i,"FTR" ]=="H",0,1))

    date <- df_raw[(df_raw$Date < df_raw[i,"Date"]),]
    home <- date[(date$HomeTeam==df_raw[i,"HomeTeam"]),]
    away <- date[(date$AwayTeam==df_raw[i,"AwayTeam"]),]



    # solo quiero los tres ultimos partidos
    if (nrow(home)>3){
      home <- tail(home, 3)
    }
    if (nrow(away)>3){
      away <- tail(away, 3)
    }

    # para los primeros partidos ( no hay suficientes para la media)
    if (dim(home)[1] == 0) {
      home <- df_raw[i,]
    }
    if (dim(away)[1] == 0) {
      away <- df_raw[i,]
    }


    df_raw[i,"FTHG_avg" ] <- mean(home[,"FTHG"])
    df_raw[i,"FTAG_avg" ] <- mean(away[,"FTAG"])

    df_raw[i,"HTHG_avg" ] <- mean(home[,"HTHG"])
    df_raw[i,"HTAG_avg" ] <- mean(away[,"HTAG"])

    df_raw[i,"HS_avg" ] <- mean(home[,"HS"])
    df_raw[i,"AS_avg" ] <- mean(away[,"AS"])

    df_raw[i,"HST_avg" ] <- mean(home[,"HST"])
    df_raw[i,"AST_avg" ] <- mean(away[,"AST"])

    df_raw[i,"HF_avg" ] <- mean(home[,"HF"])
    df_raw[i,"AF_avg" ] <- mean(away[,"AF"])

    df_raw[i,"HC_avg" ] <- mean(home[,"HC"])
    df_raw[i,"AC_avg" ] <- mean(away[,"AC"])

    df_raw[i,"HY_avg" ] <- mean(home[,"HY"])
    df_raw[i,"AY_avg" ] <- mean(away[,"AY"])

    df_raw[i,"HR_avg" ] <- mean(home[,"HR"])
    df_raw[i,"AR_avg" ] <- mean(away[,"AR"])

    df_raw[i,"HP_avg" ] <- mean(home[,"HP"])
    df_raw[i,"AP_avg" ] <- mean(away[,"AP"])


  }
  return(df_raw)

}

liga <- get_last_3()
```

Variables finales

De tal forma los datos empleados son los siguientes, que extraemos de los partidos de La Liga de las temporadas de 2010 a 2011:

Nombre	Descripción	Tipo
Season	Temporada en la que se jugó el partido	Categórica
HomeTeam	Equipo local	Categórica
AwayTeam	Equipo visitante	Categórica
Date	Fecha en la que se jugó el partido	Date

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

FTHG_avg	Media de los goles marcados por el equipo local en los tres últimos partidos	Numeric
FTAG_avg	Media de los goles marcados por el equipo visitante en los tres últimos partidos	Numeric
FTR	Resultado final	Categórica (H= Home win, D=Draw, A=Away Team)
HTHG_avg	Media de los goles marcados al descanso por el equipo local en los tres últimos partidos	Numeric
HTAG_avg	Media de los goles marcados al descanso por el equipo visitante en los tres últimos partidos	Numeric
HS_avg	Media de los tiros realizados por el equipo local en los tres últimos partidos	Numeric
AS_avg	Media de los tiros realizados por el equipo visitante en los tres últimos partidos	Numeric
HST_avg	Media de los tiros a puerta realizados por el equipo local en los tres últimos partidos	Numeric
AST_avg	Media de los tiros a puerta realizados por el equipo visitante en los tres últimos partidos	Numeric

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

HF_avg	Media de las faltas cometidas por el equipo local en los tres últimos partidos	Numeric
AF_avg	Media de las faltas cometidas por el equipo visitante en los tres últimos partidos	Numeric
HC_avg	Media de los córneres sacados por el equipo local en los tres últimos partidos	Numeric
AC_avg	Media de los córneres sacados por el equipo visitante en los tres últimos partidos	Numeric
HY_avg	Media de las tarjetas amarillas recibidas por el equipo local en los tres últimos partidos	Numeric
AY_avg	Media de las tarjetas amarillas recibidas por el equipo visitante en los tres últimos partidos	Numeric
HR_avg	Media de las tarjetas rojas recibidas por el equipo local en los tres últimos partidos	Numeric
AR_avg	Media de las tarjetas rojas recibidas por el equipo visitante en los tres últimos partidos	Numeric
B365H	Cuota de ganar el equipo local de la casa de apuestas Bet365	Numeric
B365A	Cuota de ganar el equipo visitante de la casa de apuestas Bet365	Numeric

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

B365D	Cuota de empate de la casa de apuestas Bet and Win	Numeric

Partición de datos

Para la partición de datos decidimos usar las primeras nueve temporadas para el conjunto de entrenamiento y la última como test. Nos ha parecido que de esta manera planteamos un problema de machine learning más realista, pues usamos los datos históricos para predecir los recientes. Así, la mayoría de los datos se encuentran en el conjunto de entrenamiento. Esto puede ser beneficioso para en un futuro, poder predecir temporadas venideras sólo con añadirlas al conjunto de test.

No obstante, aunque en el conjunto de entrenamiento se encuentren la mayoría de los datos, hemos comprobado que los datos de test y entrenamiento guardan una proporción similar en cuanto la clasificación de las clases.

```
########## para predecir la ultima temporada

liga_test <- subset(liga, Season=="2018-2019" )
liga_train <- subset(liga, Season !="2018-2019" )
```

```
> table(liga_train$FTR)/nrow(liga_train)

        A         D         H
0.2816613 0.2328166 0.4855221
> table(liga_test$FTR)/nrow(liga_test)

        A         D         H
0.2684211 0.2894737 0.4421053
>
```

El resultado muestra que el conjunto de test guarda una proporción de clases similar al de entrenamiento

Clasificación

La variable objetivo a estudiar es FTR (Full Time Result), que puede tomar tres posibles valores: “H” (gana local), “A” (gana visitante) y “D” (empate).

Para abordar el problema de clasificación, he decidido utilizar los siguientes modelos de aprendizaje, para compararlos y evaluar cual es mejor a la hora de predecir el resultado final de un partido.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Regresión logística múltiple: de manera similar a la regresión logística simple, aborda el problema de clasificación múltiple. Aunque este método no ha sido estudiado en clase, me ha parecido interesante crear un modelo sencillo que arroje información sobre la importancia de las variables.

KNN: se ha utilizado únicamente con las variables numéricas, por lo tanto, no teniendo en cuenta los equipos que juegan, ni la temporada en la que nos encontramos. Puede arrojar información sobre cómo de importantes son estas variables que no utilizamos.

KMeans: también utilizando las variables numéricas, he abordado el problema de clasificación por el método de KMeans.

C4.5: los árboles de decisión aportan interpretabilidad y son buenos para trabajar con datos complejos. Además, me ha parecido interesante comparar la actuación de un único árbol frente a un método de ensemble.

Random Forest: como ya sabemos, se trata de un método de bagging. Suelen ser buenos clasificadores al utilizar múltiples weak learners y es uno de los métodos más empleados.

Support Vector Machine: hemos querido aprovechar la flexibilidad de la clasificación no lineal de las SVM para encontrar un modelo óptimo.

Neural Network: finalmente hemos decidido atacar el problema con una red neuronal para buscar patrones ocultos y correlaciones entre los datos, aprovechando que la red neuronal está en continuo aprendizaje.

Regresión logística múltiple

Aunque la regresión logística permite clasificar, se trata de un modelo de regresión que modela el logaritmo de la probabilidad de pertenecer a cada grupo. La asignación final se hace en función de las probabilidades predichas.

En primer lugar, realizamos un modelo de regresión logística que comprenda todas las veriables utilizando la función multinom.

El resultado de este modelo aplicado al conjunto de train es el siguiente:

```
> tab <- table(liga_train$FTR, predictions_mn)
> tab
   predictions_mn
      A    D    H
  A 470   70  423
  D 197  105  494
  H 178   77 1405
> accuracy <- round((sum(diag(tab))/sum(tab))*100,2)
> accuracy
[1] 57.91
>
```

Para comprobar su efectividad, lo aplicamos al conjunto de test y obtenemos la siguiente matriz:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> tab <- table(liga_test$FTR, predictions_mn)
> tab
   predictions_mn
      A   D   H
  A  25   3  74
  D  13   6  91
  H   9   4 155
> accuracy <- round((sum(diag(tab))/sum(tab))*100,2)
> accuracy
[1] 48.95
>
```

Solo con los importantes

```
liga_mn <- multinom(FTR ~ FTHG_avg+FTAG_avg+HST_avg+AST_avg+HP_avg+AP_avg+B365A+B365H+B365D+HR_avg+AR_avg, data=liga_train)
summary(liga_mn)
coefs <- coef(liga_mn)
coefs
```

Ahora bien, tenemos un modelo general utilizando todas las variables como predictoras. Vamos a utilizar métodos de regularización, que consiste en ajustar el modelo incluyendo todos los predictores, pero aplicando una penalización que fuerce a que las estimaciones de los coeficientes de regresión tiendan a cero. Con esto se intenta evitar overfitting, reducir varianza, atenuar el efecto de la correlación entre predictores y minimizar la influencia en el modelo de los predictores menos relevantes. Por lo general, aplicando regularización se consigue modelos con mayor poder predictivo. Hemos estudiado dos métodos de regularización, lasso y ridge, además de una técnica para combinarlos y elegir el mejor modelo posible.

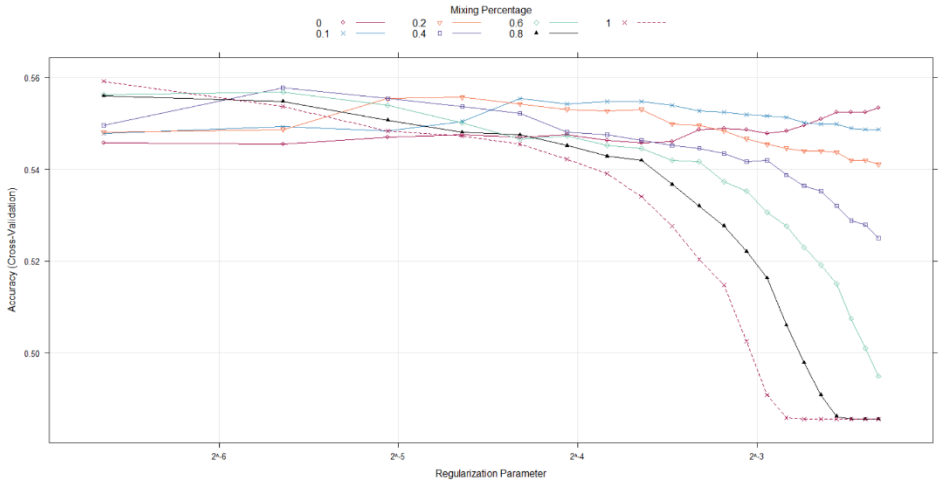
El parámetro lambda es el que regula cuán restrictivo es nuestro modelo, y alpha es un parámetro que nos indica el nivel de mezcla entre ridge y lasso, siendo 0 una regularización tipo ridge y 1, una tipo lasso.

Entrenamos nuestro modelo con el conjunto de entrenamiento, para los distintos valores de alpha y lamda.

```
#This variation of alpha is called mixing percentage
glmnet_grid <- expand.grid(alpha = c(0,  .1,  .2, .4, .6, .8, 1),
                           lambda = seq(.01, .2, length = 20))

# CV method using 5 folds
glmnet_ctrl <- trainControl(method = "cv", number = 5)
glmnet_fit <- train(FTR ~ ., data = liga_train,
                    method = "glmnet",
                    preProcess = c("center", "scale"),
                    tuneGrid = glmnet_grid,
                    trControl = glmnet_ctrl)

glmnet_fit
```

Visualizamos la precisión de nuestro clasificador en función de alpha y lambda y obtenemos el mejor modelo, que en nuestro caso utiliza Alpha 1 y lambda 0.01. Es decir, utiliza una regularización tipo lasso pura.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT




```
> glmnet_fit$bestTune
    alpha lambda
121     1   0.01
>
```

Probamos el mejor modelo en el conjunto de entrenamiento, con una precisión del 56,6%.

```
> confusionMatrix(liga_train$FTR, pred_classes)
Confusion Matrix and Statistics

          Reference
Prediction    A    D    H
         A  366    2  595
         D  129   14  653
         H  100    5 1555

Overall Statistics

               Accuracy : 0.566
                 95% CI : (0.5491, 0.5827)
    No Information Rate : 0.8198
    P-Value [Acc > NIR] : 1

                  Kappa : 0.213
```

Aplicando el modelo al conjunto de test, obtenemos una precisión de 48,42%.

Cabe destacar que, aunque el modelo predice con muy poco error las victorias de los equipos locales, funciona peor para clasificar los empates y las victorias del equipo visitante.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
Confusion Matrix and Statistics

          Reference
Prediction   A   D   H
         A  29   1  72
         D  18   3  89
         H  14   2 152

Overall Statistics

               Accuracy : 0.4842
                 95% CI : (0.4329, 0.5357)
    No Information Rate : 0.8237
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1231
```

KNN

KNN es un algoritmo de aprendizaje supervisado. Su funcionamiento es sencillo. Calcula las distancias de una observación nueva a todas las observaciones del conjunto de entrenamiento y escoge la clase mayoritaria entre los K vecinos más cercanos.

El modelo, por lo tanto, se crea utilizando conjuntamente

Algunas de las ventajas de este algoritmo es que es fácil de implementar, funciona bien en datasets pequeños y no se necesita re-entrenar el modelo si se añaden nuevos datos al conjunto de entrenamiento. Por otro lado, es un algoritmo "lazy learner" y como debe calcular la distancia de cada punto a cada dato puede llegar a ser lento.

Para este algoritmo, utilizamos únicamente las variables numéricas, pues es necesario calcular las distancias.

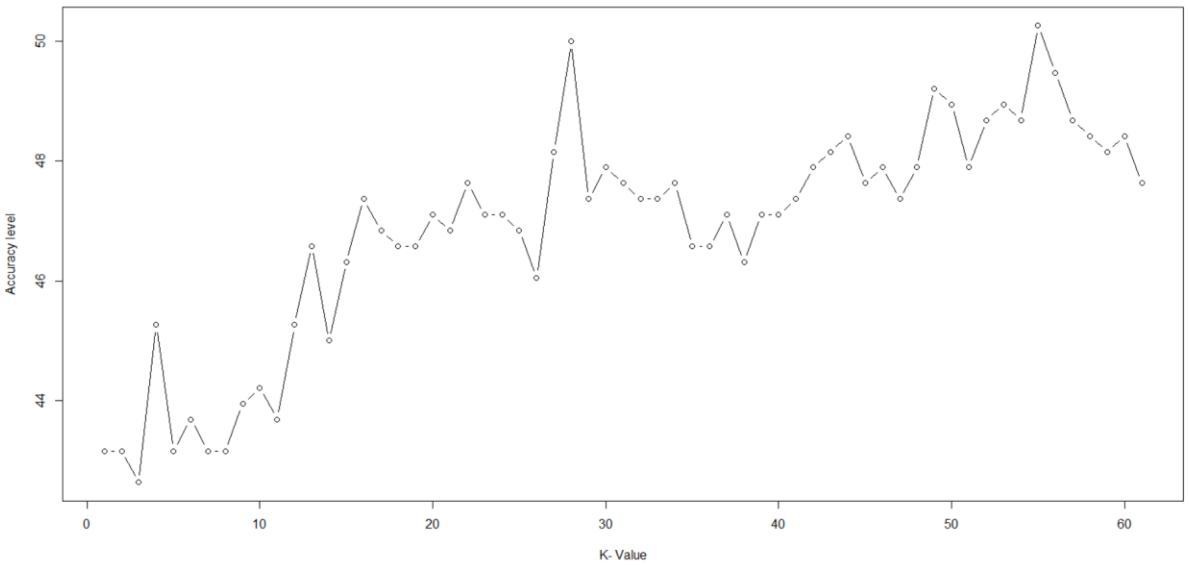
Creamos un modelo genérico, siendo k la raíz cuadrada del número total de observaciones.

Utilizando el algoritmo con k= 61, tenemos:

```
> table <-table(knn.61, test.labels)
> table
      test.labels
knn.61   A   D   H
     A  32  17  17
     D   4   4   4
     H  66  89 147
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 48.16
>
```

Una vez tenemos el modelo genérico, vamos a encontrar el mejor modelo. Para ello, debemos obtener el número óptimo de vecinos. Creamos un bucle que guarde el valor de k y la precisión del modelo para cada valor de k y representamos gráficamente el resultado.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Observamos que encontramos la mayor precisión para k= 43. Por tanto, elegimos k= 43.

```
> k.optm
[1] 55
>
```

El resultado que obtenemos en este modelo es el siguiente:

```
> table <- table(knn.optimal, test.labels)
> table
           test.labels
knn.optimal   A   D   H
          A  34  16  17
          D   5   7   4
          H  63  87 147
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 49.47
>
```

KMeans

K-Means es un algoritmo de clasificación no supervisada, que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o clúster.

Dado un número de clústeres, K, el algoritmo K-Means se realiza en tres pasos:

1) Se asignan centroides aleatorios

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

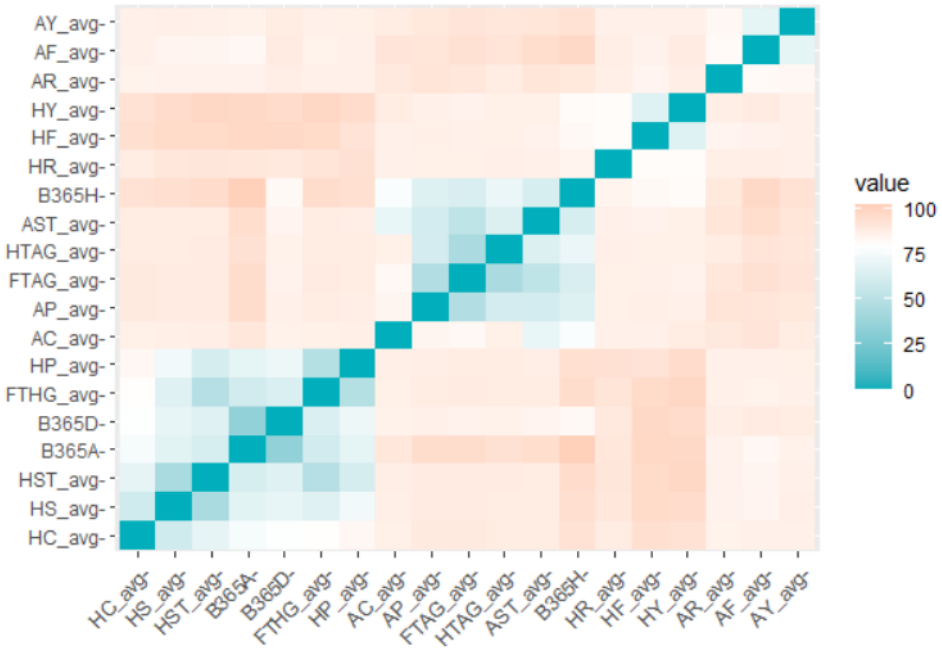
2) Se asigna cada observación al clúster con el centroide más cercano, midiendo la distancia entre la observación y todos los centroides.

3) Se generan nuevos centroides para los clústeres, siendo el centroide el centro, el punto medio del clúster.

4) Se vuelve al punto 2, y se repite el proceso hasta que no haya nuevas asignaciones (los miembros pertenecientes a cada clúster no cambian)

En primer lugar, escalamos los datos y creamos una matriz de distancias, en la que observamos que variables se encuentran a más y menos distancia. No nos sorprende encontrar que entre las variables que más distancia observamos son las odds de que gane visitante(B365A) y las odds de que gane local (B365H)

Matriz de distancias

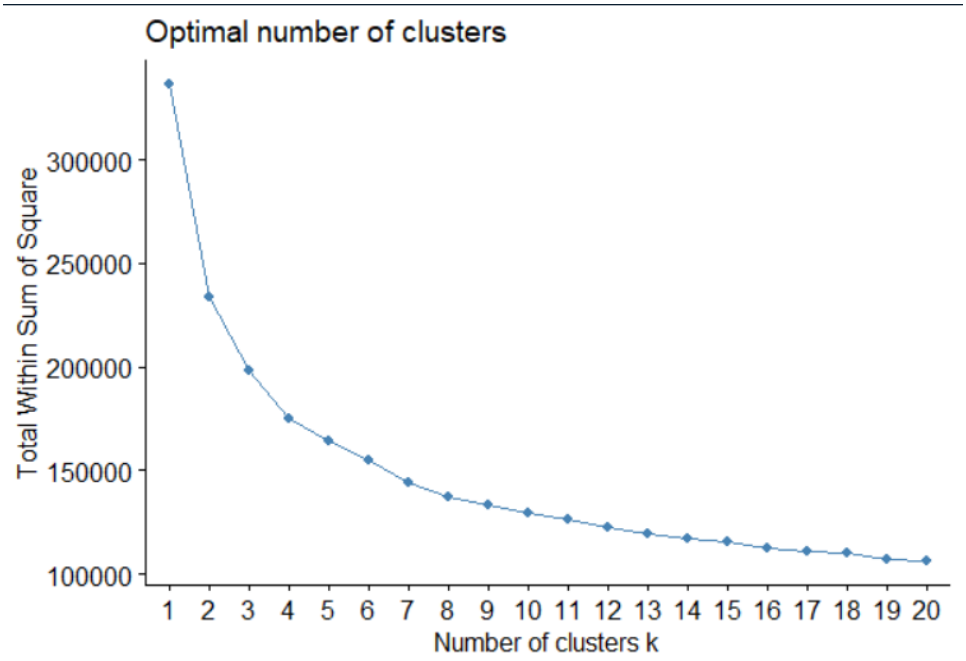


Antes de empezar con el algoritmo, debemos encontrar el número óptimo de clústeres. Para ello, empleamos tres métodos diferentes:

Elbow Method

Se trata de minimizar la suma de las variaciones dentro de los clústeres, que mide cómo de compactos son los clústers.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

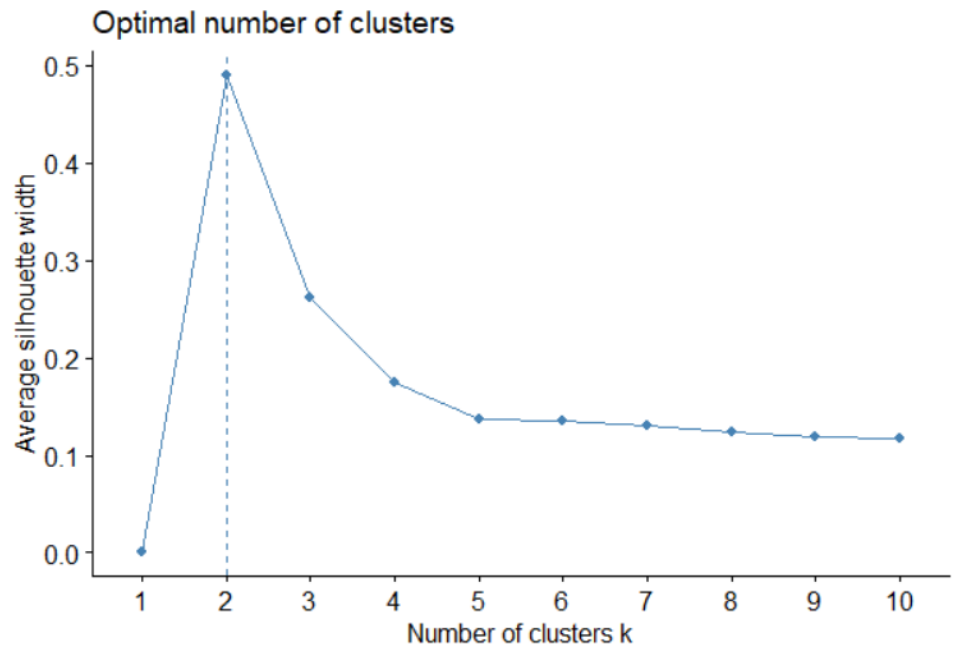


El punto óptimo es en el que encontramos un codo o "elbow". En este caso, el Método Elbow nos sugiere que utilicemos 3, 4 o 5 clústeres en nuestro problema de clasificación

Siluette method

Este método mide la calidad de una partición en clúster. Determina cuan bien se encuentra cada objeto dentro de su grupo.

Para ello, mide la distancia de cada objeto a todos los elementos de su clúster, y la distancia de cada objeto al resto de objetos que no pertenecen al clúster.

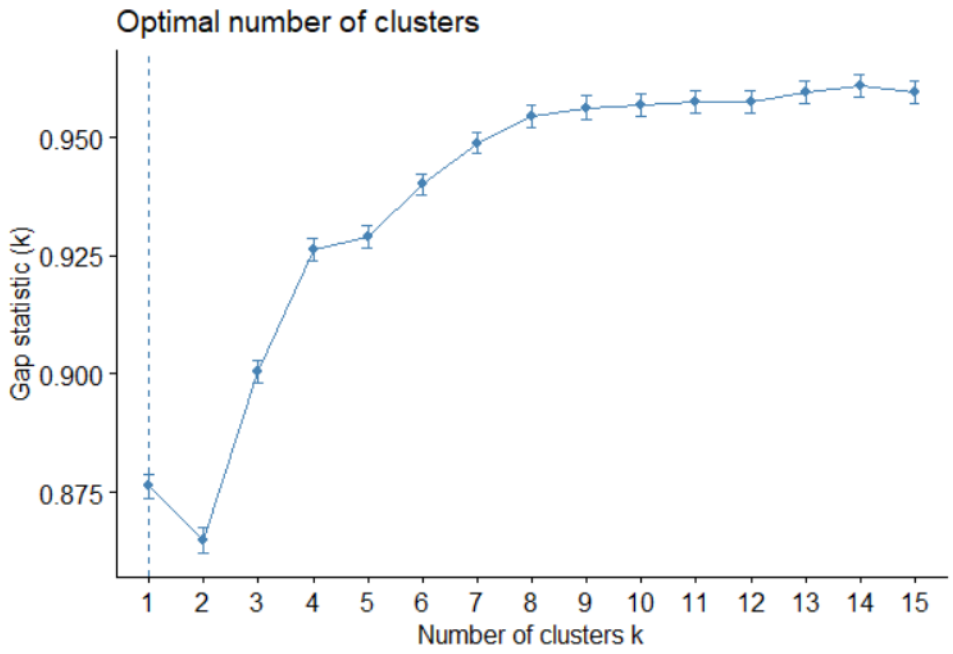


Según observamos en la imagen, el método de la silueta nos sugiere que utilicemos dos clusters.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Gap Statistic Method

El método Gap Statistic compara la variación intragrupo total para diferentes valores de k con sus valores esperados bajo nula distribución de referencia de los datos.



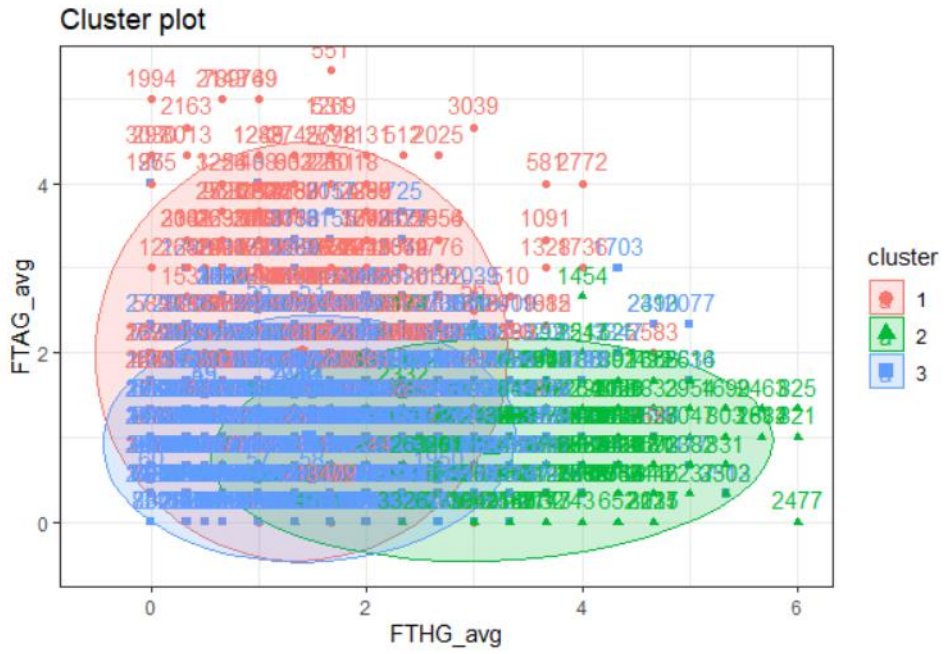
Este método nos dice que el mejor valor de k es 2.

Análisis de KMeans para k=3

Hemos seleccionado dos clústeres pues la variable FTR tiene tres posibles valores (H, A, D). Además, el método Elbow nos recomienda este valor.

Después de entrenar el modelo, visualizamos los clústeres en función de FTHG_avg y FTAG_avg. Nos ha parecido coherente, pues estas variables representan los goles que ha marcado cada equipo en los últimos tres partidos.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Una vez entrenado el modelo, lo probamos en el conjunto de entrenamiento, obteniendo el siguiente resultado:

```
> table_pred<-table(train.labels, prediction)
> table_pred
            prediction
train.labels   1   2   3
           A 566  90 307
           D 524 139 133
           H 925 600 135
>
```

En cada clúster, la clase mayoritaria es la representativa de cada clúster. Por lo tanto,

accuracy = 53,5%.

Aplicando el modelo al conjunto de test, obtenemos lo siguiente:

```
> table_pred<-table(test.labels, prediction)
> table_pred
           prediction
test.labels   1   2   3
          A  63   8  31
          D  81  15  14
          H 107  49  12
```

Tenemos una precisión en test de 49,2%

No obstante, hemos de notar que no tenemos ningún clúster con clase mayoritaria D, por lo que "perderíamos" la clase D en favor de ganar precisión.

C4.5

C4.5 es un algoritmo de árboles de decisión que mejora parte de las deficiencias encontradas en el algoritmo ID3, pues selecciona el mejor separador sin el sesgo de ganancia que afecta a ID3, además de utilizar un mecanismo de poda. Es por eso, que no hemos decidido emplear el algoritmo C4.5 en lugar del ID3.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

En C4.5, la elección del mejor separador se basa en la mayor ratio de ganancia.

Implementamos un modelo general, sin poda, obteniendo el siguiente resultado en train:

```
Evaluation on training data (3419 cases):

           Decision Tree
         ----------------
         Size      Errors

          525  608(17.8%)   <<


          (a)   (b)   (c)    <-classified as
         ----  ----  ----
          832    38    93    (a): class A
          134   529   133    (b): class D
          117    93  1450    (c): class H


        Attribute usage:

        100.00% AwayTeam
         90.17% B365D
```

Aunque tengamos un buen resultado en train, aplicando el modelo a test tenemos una precisón mucho menor, incurriendo en overfitting.

```
> predictions <- predict(tree_result, newdata = liga_test, type ="class")
> table<-table(prediction=predictions, real= liga_test$FTR)
> table
          real
prediction  A  D  H
         A 44 36 49
         D 10 15 29
         H 48 59 90
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 39.21
>
```

Para encontrar el mejor modelo y reducir overfitting, buscamos el valor óptimo del parámetro de coste. Para ello, hemos implementado un bucle que calcula el error para valor de coste entre 0.01 y 1.

```
i=0.1
suggested.i <- 100
for (i in 1:suggested.k){
  tree_result <- C5.0(FTR  ~ ., data=liga_train, rules = TRUE,
                      control = C5.0Control(
                        noGlobalPruning = FALSE, # Pruning is in effect
                        CF= i/100))   #Higher CF less prunning
  predictions <- predict(tree_result, newdata = liga_test, type ="class")

  table<-table(prediction=predictions, real= liga_test$FTR)

  error_classification <- mean(predictions != liga_test$FTR)

  cat(i/100,'=',error_classification,'\n')

}
```

Encontramos el menor error para CF = 0.2

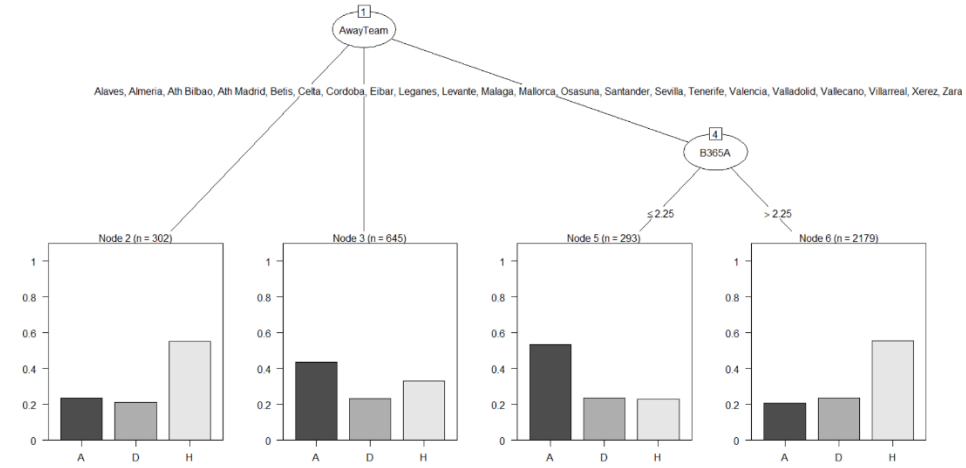
```
0.19 = 0.5157895
0.2 = 0.5157895
0.21 = 0.5210526
0.22 = 0.5210526
0.23 = 0.5210526
0.24 = 0.5210526
0.25 = 0.5210526
0.26 = 0.5289474
0.27 = 0.5289474
```

Creamos el árbol para CF=0.2:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
library(C50)
tree_result <- C5.0(FTR  ~ ., data=liga_train, rules = TRUE,
                    control = C5.0Control(
                      noGlobalPruning = FALSE, # Pruning is in effect
                      CF= 0.2))  #Higher CF less prunning
```

Observamos un árbol muy sencillo, con tan solo 4 nodos terminales.



Las reglas de decisión del árbol son:

```
Rules:

Rule 1: (342/114, lift 2.4)
        AwayTeam in {Barcelona, Real Madrid}
        ->  class A  [0.666]

Rule 2: (620/240, lift 2.2)
        B365A <= 2.25
        ->  class A  [0.613]

Rule 3: (2770/1225, lift 1.1)
        AwayTeam in {Alaves, Almeria, Ath Bilbao, Ath Madrid, Betis, Celta,
                     Cordoba, Eibar, Elche, Espanol, Getafe, Girona, Granada,
                     Hercules, La Coruna, Las Palmas, Leganes, Levante, Malaga,
                     Mallorca, Osasuna, Santander, Sevilla, Sociedad, Sp Gijon,
                     Tenerife, Valencia, Valladolid, Vallecano, Villarreal,
                     Xerez, Zaragoza}
        B365A > 2.25
```

Resaltamos que cuando el equipo visitante es Real Madrid o Barcelona, el árbol clasifica como victoria visitante, al igual que ocurre cuando las odds de Bet365 de que gane el visitante son menores de 2,25.

Aplicando el modelo al conjunto de entrenamiento obtenemos una precisión de 56.71%

```
> table
          real
prediction   A   D   H
         A 394 140 115
         D   0   0   0
         H 569 656 1545
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 56.71
>
```

El resultado de aplicar el modelo al conjunto de test es el siguiente:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> table<-table(prediction=predictions, real= liga_test$FTR)
> table
          real
prediction   A   D   H
         A  34  23  18
         D   0   0   0
         H  68  87 150
> error_classification <- mean(predictions != liga_test$FTR)
> error_classification
[1] 0.5157895
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 48.42
>
```

Aunque aumentamos la precisión considerablemente, vuelve a ocurrir que ganamos precisión a cambio de perder la clase "D".

Random Forest

Random forest es una técnica de ensemble , que surge como una mejora del proceso de bagging para prevenir la correlación entre los árboles generados con el modelo. Si un predictor tiene mucha influencia sobre el modelo, mediante el bagging siempre era seleccionado. Como consecuencia, había una gran correlación entre los árboles creados y, por tanto, bagging no conseguía reducir la varianza ni tampoco mejorar el modelo.

Sin embargo, Radom Forest prueba en cada iteración n predictores al azar, por lo que eliminamos el problema de utilizar siempre el mejor predictor.

Los hiperparámetros principales que utiliza Random Forest son:

- El número de variables que se utiliza en cada iteración
- El número óptimo de árboles (o iteraciones)
- El número mínimo de observaciones que deben tener los nodos terminales

En primer lugar, creamos un modelo general con parámetros al azar. Elegimos 500 árboles, 3 predictores por cada iteración y no especificamos el número mínimo de observaciones de los nodos terminales.

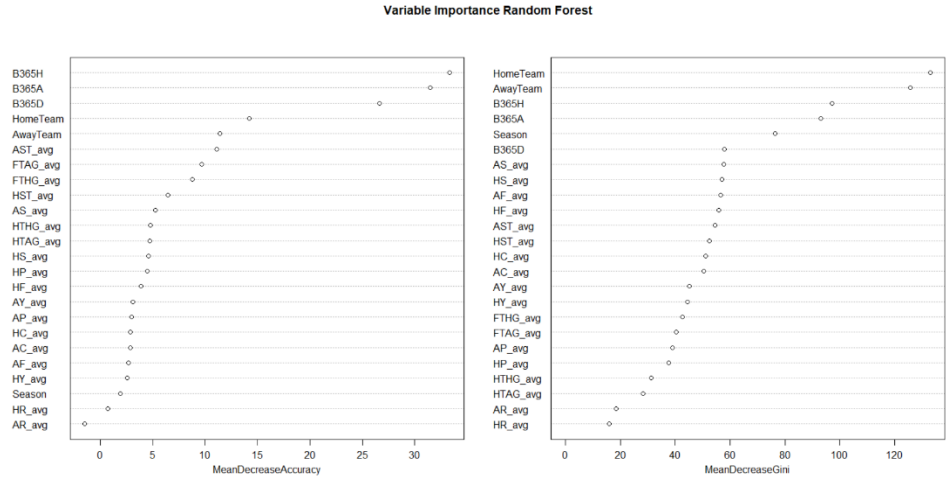
```
liga_rf <- randomForest(formula = (FTR) ~ .,data = liga_train,ntree = 500, mtry = 3,
                        importance = TRUE,na.action=na.roughfix, replace = FALSE)
```

Obtenemos el siguiente resultado:

```
Call:
 randomForest(formula = (FTR) ~ ., data = liga_train, ntree = 500,      mtry = 3, importance = TRUE, replace = FALSE, na.action = na.roug
hfix)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 44.28%
Confusion matrix:
    A  D    H class.error
A 425 37  501   0.5586708
D 191 47  558   0.9409548
H 183 44 1433   0.1367470
```

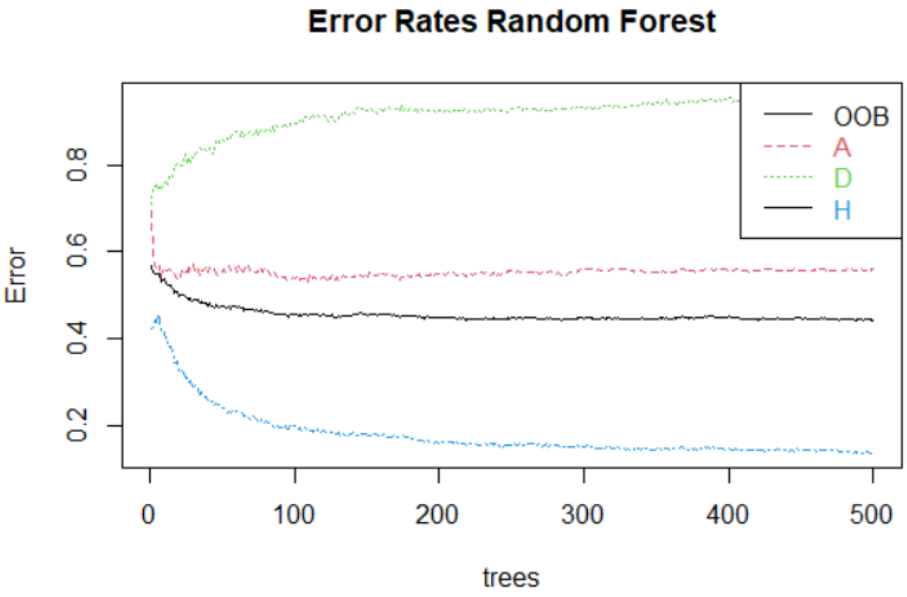
Podemos ver la influencia que tienen los diferentes predictores en el árbol, observando que los más influyentes son: El equipo local y el visitante, la temporada y las odds de las casas de apuestas.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



El error cometido para cada clase en función del número de árboles es el siguiente:

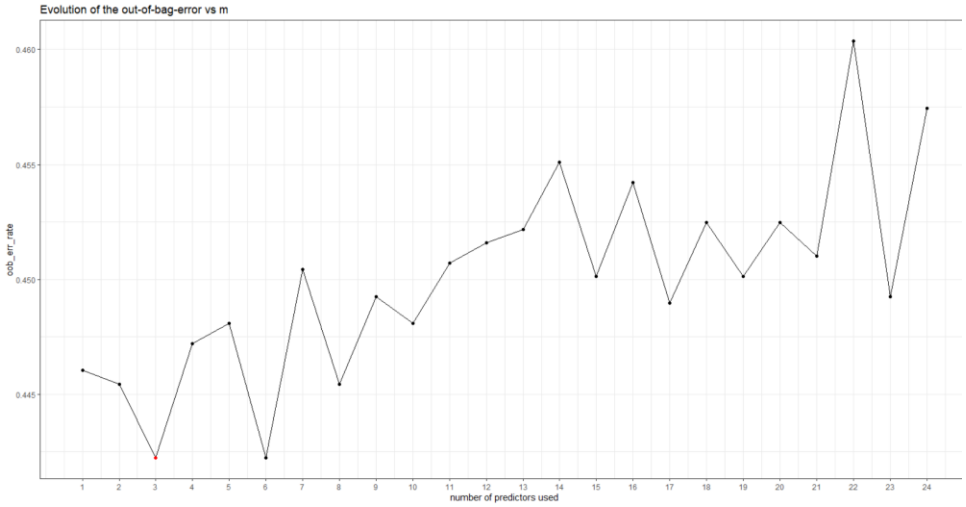
Podemos ver que, para el empate, el error aumenta a medida que aumenta el número de árboles, al contrario que para la clase victoria del local.



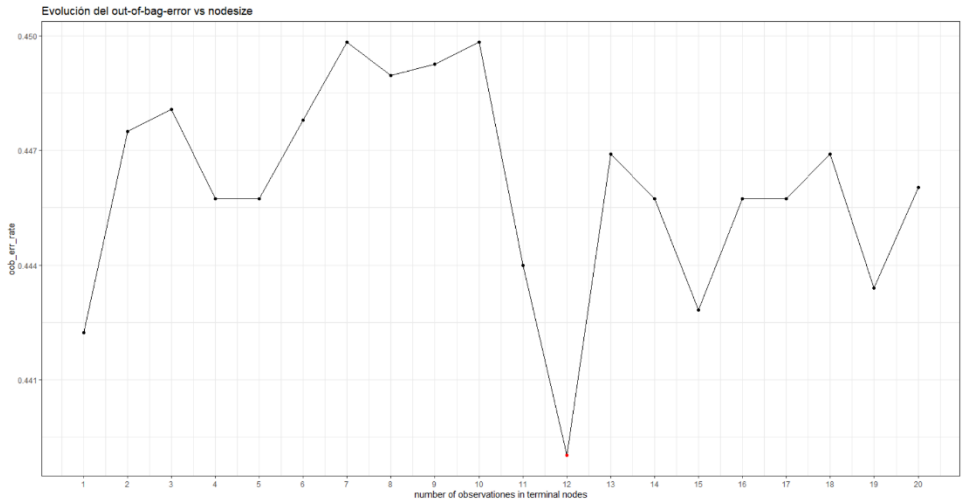
Une vez tenemos el modelo general, vamos a buscar los hiperparámetros óptimos para encontrar el mejor modelo.

En primer lugar, el número de variables que el algoritmo prueba en cada nodo. En la gráfica observamos que el valor del parámetro que minimiza el out-of-bag-error es 3.

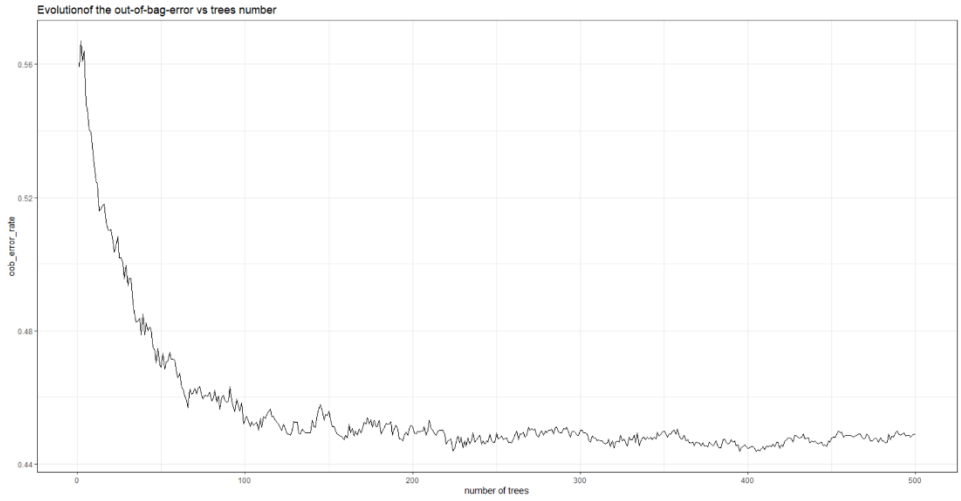
María Oliva Calero
Ignacio García Santamaría
3ºB GITT



En cuanto al número mínimo de observaciones por nodo, el valor que minimiza el error es 12.



Finalmente, vemos la evolución del error frente al número de árboles. A partir de 300 se estabiliza el error, por lo que vamos a usar este valor.

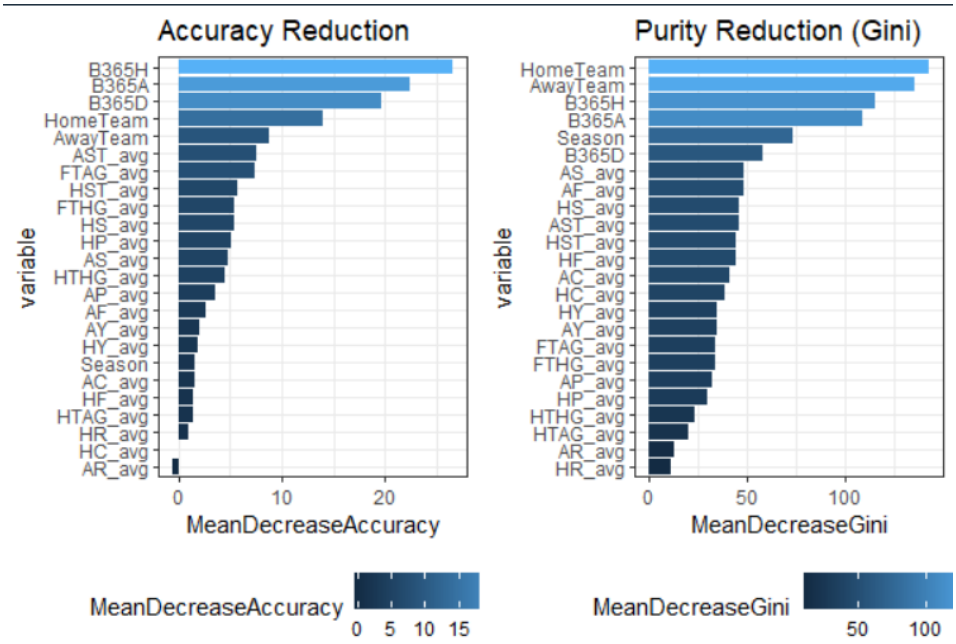


Creamos el modelo con mtry=3, ntree=300 y nodesize=12.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
Call:
 randomForest(formula = FTR ~ ., data = liga_train, mtry = 3,      ntree = 300, nodesize = 12, importance = TRUE, norm.votes = TRUE)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 3

        OOB estimate of  error rate: 44.52%
Confusion matrix:
    A  D    H class.error
A 457 43  463   0.5254413
D 216 44  536   0.9447236
H 209 55 1396   0.1590361
> |
```

Observamos la influencia de los predictores en el árbol: los más influyentes son: el equipo local y el visitante, la temporada y las odds de las casas de apuestas de nuevo.



Utilizamos el modelo de entrenamiento para ver su precisión y conseguimos un error muy pequeño. Además, observamos que el problema que teníamos anteriormente con que los modelos no detectaban correctamente los empates se ha minimizado considerablemente.

```
Confusion Matrix and Statistics

          Reference
Prediction    A    D    H
         A  903   43   38
         D    0  686    0
         H   60   67 1622

Overall Statistics

               Accuracy : 0.9392
                 95% CI : (0.9306, 0.9469)
```

Probando el modelo en test, vemos que tenemos overfitting. La precisión baja considerablemente y vuelve a aparecer el problema con el empate.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> rfcm <- confusionMatrix(ligarf_pr, liga_test$FTR)
> rfcm
Confusion Matrix and Statistics

          Reference
Prediction   A   D   H
         A  42  34  22
         D   2   1   3
         H  58  75 143

Overall Statistics

               Accuracy : 0.4895
                 95% CI : (0.4381, 0.541)
    No Information Rate : 0.4421
```

Neuronal network

Las redes neuronales son una serie de algoritmos que imitan las operaciones de un cerebro humano para reconocer las relaciones entre grandes cantidades de datos.

Entre sus ventajas encontramos:

- Auto organización: Una RNA crea su propia representación de la información en su interior.
- Tolerancia a fallos: Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente.
- Flexibilidad: Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada

Para crear el algoritmo de Neural Network hemos empleado el paquete nnet, que, aunque solo crea una capa oculta en la neurona, es óptimo para conjuntos grandes de datos.

Los parámetros que hemos utilizado para crear el modelo son:

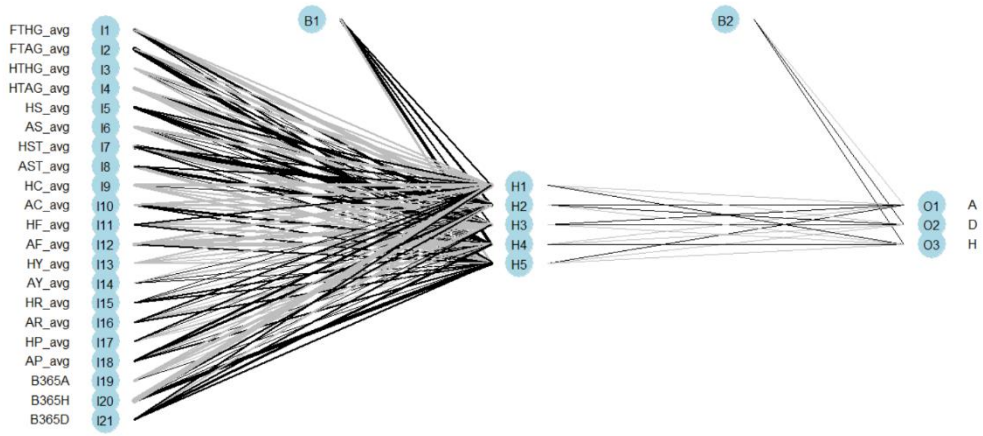
- Dimensión de la capa oculta
- Weight decay
- Maximo número de iteraciones

Finalmente, tras probar varias combinaciones hemos concluido que el mejor resultado se producía para los valores de dimensión 5, decay 0.0001 y maximo de iteraciones 1000.

El modelo final queda así:

```
model<-nnet(FTR~.,data=liga_train_num,size = 5,decay = 0.0001, abstol = 1.0e-4, reltol = 1.0e-8,maxit = 1000)
```

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Probamos el modelo en el conjunto de entrenamiento y obtenemos una precisión de 58,79%

```
> confusionMatrix(mtab)
Confusion Matrix and Statistics

          prediction
             A    D    H
  A  555   38  370
  D  250   55  491
  H  205   55 1400

Overall Statistics

               Accuracy : 0.5879
                 95% CI : (0.5712, 0.6045)
    No Information Rate : 0.6613
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2963

 Mcnemar's Test P-Value : <2e-16
```

En el conjunto de test, la precisión es de 49,47%. Es notable que, aunque el algoritmo no es preciso en cuanto a los empates, sí mejora considerablemente esta precisión frente a los modelos anteriores.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> confusionMatrix(mtab)
Confusion Matrix and Statistics

          prediction
             A   D   H
        A   39  14  49
        D   23  21  66
        H   30  10 128

Overall Statistics

                      Accuracy : 0.4947
                        95% CI : (0.4434, 0.5462)
           No Information Rate : 0.6395
           P-Value [Acc > NIR] : 1
```

Support Vector Machine

La función principal de un SVM es encontrar un hiperplano que sea capaz de separar entre dos clases. Puede haber más de un hiperplano que realice esta tarea, sin embargo, el objetivo es encontrar el hiperplano que tenga el margen máximo, es decir que la distancia entre las clases sea máxima.

Este problema se puede aplicar a la clasificación multiclase de dos maneras: one-to-all, donde cada clase se compara frente al resto y se encuentra el hiperplano que mejor realice esta separación, y one-to-one donde se comparan pares de clases hasta tener todas las combinaciones.

En primer lugar, realizo un SVM básico, con todas las variables como predictoras, para ver cómo se comporta el modelo.

```
> liga_ksvm
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0276059088907121

Number of Support Vectors : 2899

Objective Function Value : -1311.47 -1423.794 -1470.1
Training error : 0.419128
Probability model included.
>
```

Tenemos un modelo con un parámetro de coste genérico de 1 y 2899 support vectors.

La precisión de este clasificador en el conjunto de entrenamiento es la siguiente:

```
> table_tr
   prediction_tr
       A    D    H
  A  392    0  571
  D  119    2  675
  H   68    0 1592
> accuracy <- round((sum(diag(table_tr))/sum(table_tr))*100,2)
> accuracy
[1] 58.09
>
```

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Una vez que tenemos el modelo genérico, utilizamos la función tune para encontrar el mejor modelo probando distintas combinaciones de los hiperparámetros. Para encontrar el mejor modelo tune utiliza cross-validation, con cross=10.

Los hiperparámetros propios de SVM son: el coste, que regula la relación bias-variance, y gamma, que controla la curvatura de nuestro margen. Probamos los siguientes valores para los parámetros:

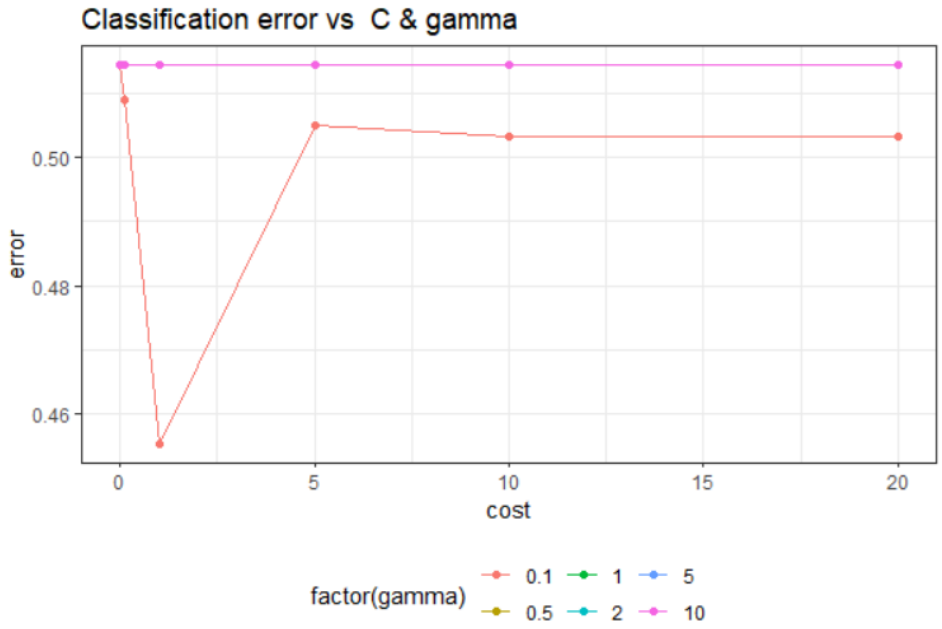
```
liga_ksmv_tune<-tune("svm", FTR~.,
                data=train,
                kernel="radial",
                ranges=list(
                  cost=c(0.01, 0.1, 1, 5, 10, 20),
                  gamma=c(0.1, 0.5, 1, 2, 5, 10)),
                tunecontrol= tune.control(sampling="cross",cross=10))

summary(liga_ksmv_tune)
```

Obteniendo el siguiente modelo:

```
> summary(liga_ksmv_tune)

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
    1   0.1

- best performance: 0.4553858

- Detailed performance results:
   cost gamma     error dispersion
1  0.01   0.1 0.5144690 0.02648782
2  0.10   0.1 0.5089117 0.02600147
3  1.00   0.1 0.4553858 0.03367311
4  5.00   0.1 0.5051182 0.01709472
```

Podemos observar la evolución del error en función de los hiperparámetros en la siguiente gráfica:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Una vez analizados los hiperparámetros, guardamos el mejor modelo que hemos obtenido con tune. El número de support vectors para nuestro mejor modelo es 3264.

```
Call:
best.tune(method = "svm", train.x = FTR ~ ., data = liga_train, ranges = list(cost = c(0.01, 0.1, 1, 5, 10,
    20), gamma = c(0.1, 0.5, 1, 2, 5, 10)), tunecontrol = tune.control(sampling = "cross", cross = 3), kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  3264
```

Probamos el modelo en nuestro conjunto de entrenamiento, obteniendo una precisión de 83,59%, las más alta hasta el momento.

```
   prediction_tr
      A    D    H
  A 745    6  212
  D  35  480  281
  H  27    0 1633
> accuracy <- round((sum(diag(table_tr))/sum(table_tr))*100,2)
> accuracy
[1] 83.59
>
```

No obstante, debemos probar el modelo en el conjunto de test:

```
> table
   prediction
      A   D   H
  A  33   2  67
  D  21   2  87
  H  13   4 151
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 48.95
>
```

La precisión, aunque buena, es mucho menor, lo que nos hace pensar que hemos incurrido en un problema de overfitting.

Comparación de los modelos de clasificación

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Finalmente, tras probar todos los algoritmos anteriores, es hora de elegir el mejor.

Para ello nos basamos en la precisión en el conjunto de test como herramienta fudamental, con algunos matices.

Para los mejores modelos de cada algotimo tenemos los siguientes resultados:

Modelo	Precisión en test
Regresión logística múltiple	48,42%
KNN	48,95%
KMeans	49,2%
C4.5	48,42%
Random Forest	48,95%
Red Neuronal	49,47%
Support Vector Machine	48,95%

Como podemos ver en la tabla superior, los resultados no varían mucho. Es algo sorprendente, pues tanto para algoritmos muy simples como KNN y otros complejos como redes neurnales, el resultado es muy similar.

Los algoritmos han conseguido casi 50% de precisión, pero teniendo en cuenta que para una predicción aleatoria, la probalidad de éxito es de 33% (Local/Empate/Visitante), es una buena señal, mostrando que hemos sido capaces de identificar algunos patrones después de todo.

Podriamos decir que nos resulta indiferente que modelo elegir, en base a la precisión, pues hay muy poca variación. No obstante, el modelo que menor overfitting presenta es el de KMeans, pues en el conjunto de entrenamiento hay una precisión de 49,2%.

Por otra parte, hemos visto que ninguno de los modelos predice correctamente la clase Empate. Aún siendo así, el algoritmo que mejor predice esta clase es Neural Network.

Por lo tanto nos encontramos ante una encrucijada a la hora de elegir el mejor modelo: ganar precisión en el empate a costa de aumentar el overfitting o perder esta precisión pero tener un modelo con menos varianza.

Hemos decidido apostar por esto último, y por lo tanto, podríamos decir que el mejor modelo para clasificar los resultados de partidos de la liga es KMeans.

Comprobación con casos específicos

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Una vez identificado el mejor modelo, creamos un conjunto de prueba nuevo, con únicamente 4 casos y predecimos el resultado, utilizando nuestro mejor modelo.

```
> table_pred
           prediction
test.labels 1 2 3
          A 0 0 0
          D 1 1 0
          H 1 0 1
> accuracy <- round((sum(diag(table_pred))/sum(table_pred))*100,2)
> accuracy
[1] 50
>
```

Para el nuevo conjunto con casos específicos obtenemos una precisión del 50%, acorde con lo esperado.

Regresión

Descripción de los módulos

En este apartado de la práctica se van a emplear diversos módulos para predecir dos variables, los goles marcados por el equipo local y los goles marcados por el equipo visitante. En muchas ocasiones, la necesidad de predecir dos variables nos va a provocar vernos obligados a realizar un modelo para cada una, no obstante, para la red neuronal, como se explicará más adelante, solo hará falta un modelo.

La división del trabajo se va a dividir en cada algoritmo empleado, mostrando los resultados obtenidos para las dos variables a predecir. Por último, en cada algoritmo se explicará su utilidad y su aportación a la práctica, además de características a resaltar de este.

El análisis de cada algoritmo se va a realizar desde dos puntos de vista. Uno para el conjunto de datos de entrenamiento y otro para el conjunto de datos de test. En ambos casos se va a ver la capacidad de predicción del modelo obteniendo el error cuadrático medio de cada predicción. Además, se hará un análisis de los residuos mediante un histograma, mediante la representación gráfica de los valores predichos y los valores reales, y mediante una comprobación de correlación.

En todos los modelos realizados no se consideró como variable ni “el resultado final” ni “los goles al descanso” porque me condicionan mucho la predicción, además de que no sería real la predicción realizada. Además, tampoco considero las variables de “temporada” ni “fecha” debido a la naturaleza del conjunto de datos de entrenamiento y de prueba.

Regresión lineal multivariante

Explicación del modelo

Este es el algoritmo más sencillo dentro de los que van a ser utilizados. Se busca la mejor combinación lineal de los parámetros de las variables explicativas a estudiar para reducir la suma del error cuadrático cometido. El motivo por el cual el primer algoritmo empleado es este es porque nos permite un buen acercamiento a las variables, ya que mediante el p-valor asociado a cada parámetro es muy fácil determinar las variables significativas.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Por otra parte, este algoritmo nos permite de manera sencilla qué variables están más relacionadas entre sí mediante la colinealidad (correlación fuerte entre las variables de entrada que puede provocar inestabilidad en el modelo y una estimación arbitraria).

Por último, el análisis en regresión lineal multivariante nos permite realizar interacciones entre las variables para así obtener una mayor flexibilidad del modelo y tener un efecto de sinergia en la salida, respetando en todo momento el principio de jerarquía.

Análisis con R

A la hora de establecer todos los modelos realizados se empleó el conjunto de datos de entrenamiento. A la hora de determinar el mejor modelo se realizó usando tanto el conjunto de entrenamiento como el conjunto de test para los diferentes modelos desarrollados. No obstante, solo se mostrará en profundidad el análisis obtenido para el mejor modelo de regresión lineal. A lo largo del análisis de este algoritmo se van a hacer referencia a los dos modelos previos al definitivo para justificar las decisiones tomadas.

Primer modelo

En primer lugar, se realizó un modelo inicial donde se buscó la determinación de las variables significativas. Para ello se analizó tanto el p-valor asociado a cada variable como los intervalos de confianza. En ambos casos se obtuvieron un gran número de coeficientes, debido al gran número de variables dummies en las variables categóricas de equipo local y de equipo visitante. Además, el modelo se consideraba significativo tanto debido a un F-estadístico bajo en ambos casos.

Equipo visitante

Para el equipo visitante se obtuvieron como variables significativas debido a su bajo p-valor las mostradas en la Ilustración inferior. Como se puede apreciar, tan solo algunas variables dummies son significativas, estas variables coinciden con aquellos equipos como el Madrid o el Barcelona (que tienen un alto porcentaje de victorias), o equipos como el Hércules o el Córdoba (que tienen un bajo porcentaje de victorias). Por otra parte, el resto de coeficientes que se corresponden con los atributos numéricos, curiosamente coinciden que los datos relacionados con el equipo visitante son los que más nos ayudan a predecir los goles anotado por este, demostrándose la correlación entre el resultado y las estadísticas del partido.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
AwayTeamAth Madrid  -7.207e-02  7.352e-02  -0.980  0.32700
AwayTeamBarcelona    2.516e-01  8.723e-02   2.885  0.00394 **
AwayTeamBetis        1.110e-02  7.431e-02   0.149  0.88130
AwayTeamCelta        1.200e-01  7.437e-02   1.613  0.10677
AwayTeamCordoba     -2.474e-01  1.240e-01  -1.996  0.04601 *
AwayTeamEibar        1.022e-01  7.854e-02   1.301  0.19326
AwayTeamElche       -4.358e-02  9.801e-02  -0.445  0.65656
AwayTeamEspanol      2.341e-03  7.099e-02   0.033  0.97369
AwayTeamGetafe      -2.143e-02  7.193e-02  -0.298  0.76578
AwayTeamGirona      -1.274e-01  9.786e-02  -1.302  0.19307
AwayTeamGranada     -1.750e-02  7.621e-02  -0.230  0.81836
AwayTeamHercules    -3.953e-01  1.244e-01  -3.176  0.00150 **
AwayTeamHuesca       1.483e-01  1.237e-01   1.199  0.23059
AwayTeamLa Coruna   -1.788e-03  7.460e-02  -0.024  0.98089
AwayTeamLas Palmas   1.449e-02  8.800e-02   0.165  0.86922
AwayTeamLeganes     -1.089e-01  8.752e-02  -1.244  0.21362
AwayTeamLevante      1.009e-01  7.286e-02   1.385  0.16611
AwayTeamMalaga      -3.592e-02  7.215e-02  -0.498  0.61862
AwayTeamMallorca    -5.889e-02  8.284e-02  -0.711  0.47720
AwayTeamOsasuna      1.139e-02  7.646e-02   0.149  0.88156
AwayTeamReal Madrid  2.694e-01  8.317e-02   3.239  0.00121 **
AwayTeamSantander    8.300e-02  8.821e-02   0.941  0.34680
AwayTeamSevilla      9.638e-02  7.239e-02   1.331  0.18312
AwayTeamSociedad     1.245e-01  7.233e-02   1.721  0.08527 .
AwayTeamSp Gijon     3.009e-02  7.869e-02   0.382  0.70221
AwayTeamTenerife     1.208e-01  1.256e-01   0.962  0.33631
AwayTeamValencia     1.004e-01  7.215e-02   1.391  0.16430
AwayTeamValladolid  -4.557e-03  8.246e-02  -0.055  0.95593
AwayTeamVallecano    7.394e-02  7.709e-02   0.959  0.33759
AwayTeamVillarreal  -1.136e-01  7.229e-02  -1.572  0.11611
AwayTeamXerez        1.475e-01  1.255e-01   1.176  0.23979
AwayTeamZaragoza    -4.716e-02  8.302e-02  -0.568  0.57003
HTRD                -2.668e-02  2.048e-02  -1.303  0.19281
HTRH                -4.148e-03  2.167e-02  -0.191  0.84824
HS_avg               1.399e-03  3.848e-03   0.364  0.71625
AS_avg              -1.913e-02  4.265e-03  -4.485 7.51e-06 ***
HST_avg              1.608e-03  7.374e-03   0.218  0.82741
AST_avg              2.085e-01  8.527e-03  24.452  < 2e-16 ***
HC_avg              -3.501e-03  5.144e-03  -0.681  0.49613
AC_avg              -3.846e-02  5.689e-03  -6.760 1.60e-11 ***
HF_avg              -1.939e-03  3.227e-03  -0.601  0.54797
AF_avg               1.716e-03  3.164e-03   0.542  0.58755
HY_avg               6.133e-03  9.771e-03   0.628  0.53022
AY_avg              -1.523e-02  9.319e-03  -1.635  0.10219
HR_avg               3.989e-03  3.816e-02   0.105  0.91676
AR_avg               3.247e-02  3.256e-02   0.997  0.31877
HP_avg              -4.099e-03  1.158e-02  -0.354  0.72333
AP_avg               4.287e-01  1.187e-02  36.116  < 2e-16 ***
B365A               -1.441e-02  6.061e-03  -2.377  0.01748 *
B365H                1.224e-03  7.774e-03   0.157  0.87491
B365D                3.781e-02  1.622e-02   2.331  0.01979 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Ilustración 1: Resumen del modelo

Para este modelo se obtuvo un R-cuadrado de 0’64 y un R-cuadrado ajustado de 0’6357. No obstante, todavía puede ser mejorado debido al gran número de variables significativas que no son útiles para la predicción de la variable en cuestión.

Equipo local

Para el equipo Local se obtuvieron como variables significativas debido a su bajo p-valor las mostradas en la Ilustración inferior. En este caso, y a diferencia del modelo anterior, se pueden considerar un mayor número de variables dummies como variables significativas. Este hecho ya no se limita a aquellos equipos con un gran número de victorias como local o como visitante. Por tanto, se puede afirmar que a la hora de predecir las los goles anotados por el equipo local, hay que prestar más atención a los equipos que están jugando, pudiendo suponerse aquí el impacto de la afición local en el resultado.

Por otra parte, el resto de coeficientes que se corresponden con los atributos numéricos, curiosamente coinciden que los datos relacionados con el equipo local son los que más nos ayudan a predecir los goles anotado por este, demostrándose la correlación entre el resultado y las estadísticas del partido.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
HomeTeamMallorca     0.1516126  0.0924745   1.640  0.10119
HomeTeamOsasuna      0.0329181  0.0849127   0.388  0.69828
HomeTeamReal Madrid  0.4620826  0.0959382   4.816 1.52e-06 ***
HomeTeamSantander   -0.0275301  0.0983221  -0.280  0.77949
HomeTeamSevilla      0.2649952  0.0808510   3.278  0.00106 **
HomeTeamSociedad     0.3466908  0.0807630   4.293 1.81e-05 ***
HomeTeamSp Gijon     0.1003974  0.0872213   1.151  0.24978
HomeTeamTenerife     0.0739165  0.1386538   0.533  0.59400
HomeTeamValencia     0.2252399  0.0806413   2.793  0.00525 **
HomeTeamValladolid   0.2046024  0.0916757   2.232  0.02569 *
HomeTeamVallecano    0.2131178  0.0849894   2.508  0.01220 *
HomeTeamVillarreal   0.1136801  0.0808820   1.406  0.15996
HomeTeamXerez        0.0983218  0.1384761   0.710  0.47773
HomeTeamZaragoza     0.1145069  0.0917233   1.248  0.21197
AwayTeamAlmeria     -0.2603870  0.0912713  -2.853  0.00436 **
AwayTeamAth Bilbao  -0.1416019  0.0790557  -1.791  0.07335 .
AwayTeamAth Madrid  -0.0219166  0.0814385  -0.269  0.78785
AwayTeamBarcelona    0.0023462  0.0966278   0.024  0.98063
AwayTeamBetis       -0.0821485  0.0823186  -0.998  0.31838
AwayTeamCelta       -0.0942695  0.0823877  -1.144  0.25261
AwayTeamCordoba     -0.2820934  0.1373134  -2.054  0.04001 *
AwayTeamEibar       -0.2123204  0.0870064  -2.440  0.01472 *
AwayTeamElche       -0.2679670  0.1085656  -2.468  0.01362 *
AwayTeamEspanol     -0.2155615  0.0786357  -2.741  0.00615 **
AwayTeamGetafe      -0.2021088  0.0796789  -2.537  0.01124 *
AwayTeamGirona      -0.2874058  0.1084009  -2.651  0.00805 **
AwayTeamGranada     -0.2209299  0.0844218  -2.617  0.00891 **
AwayTeamHercules    -0.2162154  0.1378494  -1.568  0.11685
AwayTeamHuesca      -0.2676166  0.1369971  -1.953  0.05084 .
AwayTeamLa Coruna   -0.2014524  0.0826420  -2.438  0.01483 *
AwayTeamLas Palmas  -0.1812342  0.0974776  -1.859  0.06307 .
AwayTeamLeganes     -0.1527844  0.0969488  -1.576  0.11513
AwayTeamLevante     -0.2604412  0.0807139  -3.227  0.00126 **
AwayTeamMalaga      -0.1479947  0.0799201  -1.852  0.06414 .
AwayTeamMallorca    -0.2727035  0.0917617  -2.972  0.00298 **
AwayTeamOsasuna     -0.2314731  0.0846962  -2.733  0.00631 **
AwayTeamReal Madrid -0.0300575  0.0921353  -0.326  0.74427
AwayTeamSantander   -0.1987117  0.0977168  -2.034  0.04207 *
AwayTeamSevilla     -0.1365341  0.0801907  -1.703  0.08872 .
AwayTeamSociedad    -0.1163931  0.0801193  -1.453  0.14638
AwayTeamSp Gijon    -0.2226578  0.0871640  -2.554  0.01067 *
AwayTeamTenerife    -0.2695895  0.1391842  -1.937  0.05283 .
AwayTeamValencia    -0.1193011  0.0799289  -1.493  0.13563
AwayTeamValladolid  -0.1896439  0.0913486  -2.076  0.03796 *
AwayTeamVallecano   -0.2056963  0.0853985  -2.409  0.01606 *
AwayTeamVillarreal  -0.1072159  0.0800787  -1.339  0.18069
AwayTeamXerez       -0.3496097  0.1390127  -2.515  0.01195 *
AwayTeamZaragoza    -0.2455392  0.0919648  -2.670  0.00762 **
HTRD                -0.0302928  0.0226858  -1.335  0.18185
HTRH                -0.0224104  0.0240099  -0.933  0.35068
HS_avg              -0.0328871  0.0042631  -7.714 1.55e-14 ***
AS_avg              -0.0049945  0.0047246  -1.057  0.29053
HST_avg              0.2365849  0.0081683  28.964  < 2e-16 ***
AST_avg             -0.0003533  0.0094461  -0.037  0.97016
HC_avg              -0.0326551  0.0056983  -5.731 1.08e-08 ***
AC_avg               0.0131686  0.0063023   2.089  0.03673 *
HF_avg              -0.0064816  0.0035748  -1.813  0.06989 .
AF_avg               0.0036227  0.0035054   1.033  0.30145
HY_avg              -0.0082501  0.0108233  -0.762  0.44596
AY_avg              -0.0046272  0.0103231  -0.448  0.65401
HR_avg               0.0426549  0.0422767   1.009  0.31307
AR_avg              -0.0084304  0.0360693  -0.234  0.81521
HP_avg               0.4425137  0.0128265  34.500  < 2e-16 ***
AP_avg               0.0011606  0.0131477   0.088  0.92967
B365A                0.0097147  0.0067135   1.447  0.14797
B365H               -0.0282400  0.0086120  -3.279  0.00105 **
B365D                0.0390309  0.0179649   2.173  0.02987 *
```

Ilustración 2: Resumen del modelo

Para este modelo se obtuvo un R-cuadrado de 0'70 y un R-cuadrado ajustado de 0'69. Los resultados obtenidos son similares al modelo anterior, e indican que todavía hay capacidad de mejora mediante la simplificación del modelo.

Segundo modelo

En segundo lugar, seleccionamos las variables significativas para cada uno de los modelos, basándonos en los resultados obtenidos a la hora de obtener el p-valor de coeficiente. De esta manera, se eliminan las variables redundantes que no tienen ninguna aportación al modelo. En este caso, una vez simplificado se analizó la colinealidad entre los coeficientes, indicándonos que variables estaban más relacionadas entre sí provocando un peor rendimiento del modelo.

Equipo visitante

A la hora de seleccionar las variables significativas del apartado 1, para el modelo visitante nos quedamos con: "HomeTeam", "AwayTeam", "AS_avg", "AST_avg", "AC_avg", "AP_avg", además de las variables relativas a la cuota de la casa de apuestas. Dentro de las variables significativas, salvo los

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

equipos que no eran considerados como tal en el apartado anterior, el resto de variables eran consideras como significativas, indicando una correcta elección de las mismas.

Para este modelo se obtuvo un R-cuadrado de 0'64 y un R-cuadrado ajustado de 0'63. Un modelo con una capacidad predictiva prácticamente idéntica al modelo anterior. Además, a la hora de enfrentarse a datos nuevos, el modelo había reducido el sobreajuste ya que su capacidad de predicción mejoró en pequeña medida.

Tal y como se muestra en la imagen inferior, a la hora de estudiar la colinealidad, esta se considera considerable cuando supera el valor de 10, y preocupante cuando es superior a 5. Claramente hay una alta colinealidad entre la cuota de la casa de apuestas para el equipo local y para el equipo visitante. Adicionalmente, la colinealidad entre el equipo local y el equipo visitante es preocupante. No obstante, para el tercer modelo nos desprendimos únicamente de la variable del equipo visitante, basándome como criterio de elección que tenía un mayor p-valor y un valor de coeficiente muy parecido a la variable con la que se relaciona.

```
> vif(model2_visitante)
              GVIF Df GVIF^(1/(2*Df))
HomeTeam  5.987743 34        1.026669
AwayTeam  7.042211 34        1.029121
AS_avg    3.010253  1        1.735008
AST_avg   3.018194  1        1.737295
AC_avg    1.508242  1        1.228105
AP_avg    1.651726  1        1.285195
B365A    18.916963  1        4.349364
B365H     6.674067  1        2.583422
B365D    16.463325  1        4.057502
```

Ilustración 3: Análisis de colinealidad

Equipo Local

A la hora de seleccionar las variables significativas del apartado 1, para el modelo visitante nos quedamos con: "HomeTeam", "AwayTeam", "HS_avg","HC_avg", "HP_avg", además de las variables relativas a la cuota de la casa de apuestas. Dentro de las variables significativas, salvo los equipos que no eran considerados como tal en el apartado anterior, el resto de variables eran consideras como significativas, indicando una correcta elección de las mismas. La elección de las variables fue paralela, solo cambiamos las variables del equipo local por las del visitante.

Para este modelo se obtuvo un R-cuadrado de 0'64 y un R-cuadrado ajustado de 0'63. La capacidad predictiva de mi modelo para predecir el conjunto de datos de prueba se ha visto reducida, no obstante, el sobreajuste presente en el modelo anterior se redujo, este hecho se mostrará numéricamente en el modelo definitivo.

A la hora de estudiar la colinealidad, a diferencia de en el modelo anterior, quizás porque en este caso no se escogieron todas las variables relacionadas con la casa de apuestas, no hay ningún valor para la colinealidad que sea excesivamente alto. Sigue habiendo una colinealidad preocupante entre las variables que relacionan al equipo local y al equipo visitante, no obstante, como en el caso anterior, se va a pasar por alto este hecho.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> vif(model2_local)
             GVIF Df GVIF^(1/(2*Df))
HomeTeam 7.275383 34        1.029614
AwayTeam 4.762753 34        1.023219
HS_avg   2.016046  1        1.419875
HC_avg   1.578407  1        1.256347
HP_avg   1.411808  1        1.188195
B365H    4.625166  1        2.150620
B365D    3.601348  1        1.897722
```

Ilustración 4: Análisis de colinealidad

En tercer lugar, se desarrolló un último modelo donde se eliminó la presencia de las variables que tenían colinealidad entre sí. Subsanando ese error, se realizaron las diferentes interacciones necesarias para obtener el mejor modelo en su conjunto.

Tercer modelo

En tercer lugar, se desarrolló un último modelo donde se eliminó la presencia de las variables que tenían colinealidad entre sí. Subsanado ese error, realizamos las interacciones necesarias entre variables para obtener una mejor capacidad predictiva de mi modelo.

Equipo visitante

Dentro de las interacciones realizadas en el equipo visitante, se probaron todas las combinaciones posibles. Descartándose instantáneamente las interacciones entre equipo local y equipo visitante debido al gran número de coeficientes que aparecían. Finalmente, se decidió que las interacciones que mejoraban en mayor medida tanto el rendimiento para el conjunto de entrenamiento como para el conjunto de prueba eran entre las variables “AS_avg” y “AC_avg” y entre “AS_avg” y “AP_avg”.

Adicionalmente, tal y como se muestran en las dos representaciones que se aprecian a continuación. En primer lugar, podemos comprobar que la interacción realizada ha resultado efectiva, ya que es considerada como significativa debido al bajo p-valor que tiene asociado ese coeficiente. En segundo lugar, vemos que la interacción a afectado considerablemente a los valores de la colinealidad, no obstante, se puede desconsiderar este efecto ya que esta es una de las principales consecuencias de llevar a cabo esta acción.

```
AwayTeamXerez         0.0840902  0.1233529   0.682 0.495469
AwayTeamZaragoza     -0.0691194  0.0818522  -0.844 0.398477
AS_avg                0.0004051  0.0077662   0.052 0.958407
AST_avg               0.2071055  0.0084768  24.432  < 2e-16 ***
AC_avg                0.0472353  0.0155595   3.036 0.002416 **
AP_avg                0.2649938  0.0369806   7.166 9.28e-13 ***
B365A                -0.0008818  0.0027888  -0.316 0.751873
AS_avg:AC_avg        -0.0075833  0.0012851  -5.901 3.94e-09 ***
AS_avg:AP_avg         0.0146767  0.0031401   4.674 3.06e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4618 on 3722 degrees of freedom
Multiple R-squared:  0.6471,    Adjusted R-squared:  0.6399
F-statistic: 90.98 on 75 and 3722 DF,  p-value: < 2.2e-16
```

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> vif(model3_visitante)
                    GVIF Df GVIF^(1/(2*Df))
HomeTeam        4.379662 34        1.021958
AwayTeam        3.853530 34        1.020036
AS_avg         10.289123  1        3.207666
AST_avg         3.027579  1        1.739994
AC_avg         11.478758  1        3.388032
AP_avg         16.444225  1        4.055148
B365A           4.081187  1        2.020195
AS_avg:AC_avg  26.420722  1        5.140109
AS_avg:AP_avg  24.371624  1        4.936763
```

Ilustración 5: Resumen del modelo y análisis de colinealidad

Como se puede observar, el modelo obtenido cuenta con un R-cuadrado de 0'65 y un R-cuadrado ajustado de 0'63. Este es el mejor resultado obtenido para todos los modelos realizados con esta variable. Aun así, el principal beneficio se aprecia a la hora de comprobar el rendimiento con los datos de prueba, el cual se estudiará en el siguiente apartado.

Equipo Local

Dentro de las interacciones realizadas en el equipo visitante, se probaron todas las combinaciones posibles. Descartándose del mismo modo que para el modelo anterior las variables dummies relacionadas con los equipos locales y visitantes. Finalmente, se decidió que las interacciones que mejoraban en mayor medida tanto el rendimiento para el conjunto de entrenamiento como para el conjunto de prueba eran entre las variables "AH_avg" y "AH_avg" y entre "AH_avg" y "AH_avg".

Fijándonos en las Ilustraciones que describen el comportamiento del modelo, al hacer un análisis del modelo vemos que la combinación que ha resultado más significativa ha sido la segunda interacción descrita, podríamos haber prescindido de la primera debido al alto p-valor relacionado a su coeficiente. En último lugar, al igual que en el apartado anterior, vemos que la colinealidad entre variables se ha visto afectada debido a las interacciones, la explicación es la misma que para el apartado anterior.

```
AwayTeamSevilla     -0.1910622  0.0864650  -2.210 0.027186 *
AwayTeamSociedad    -0.1445687  0.0869754  -1.662 0.096561 .
AwayTeamSp Gijon    -0.2710450  0.0952710  -2.845 0.004466 **
AwayTeamTenerife    -0.3673051  0.1509547  -2.433 0.015012 *
AwayTeamValencia    -0.1312993  0.0863857  -1.520 0.128616
AwayTeamValladolid  -0.2152179  0.1000306  -2.152 0.031499 *
AwayTeamVallecano   -0.2614480  0.0921886  -2.836 0.004593 **
AwayTeamVillarreal  -0.1461119  0.0873333  -1.673 0.094404 .
AwayTeamXerez       -0.5719558  0.1508334  -3.792 0.000152 ***
AwayTeamZaragoza    -0.3189198  0.0997480  -3.197 0.001399 **
HS_avg              -0.0075538  0.0103267  -0.731 0.464531
HC_avg              -0.0427109  0.0203398  -2.100 0.035808 *
HP_avg               0.2105892  0.0484564   4.346 1.42e-05 ***
B365H               -0.0442180  0.0078144  -5.659 1.64e-08 ***
B365D                0.0663926  0.0092120   7.207 6.88e-13 ***
HS_avg:HC_avg        0.0006411  0.0013270   0.483 0.629003
HS_avg:HP_avg        0.0266007  0.0033901   7.847 5.54e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5656 on 3722 degrees of freedom
Multiple R-squared:  0.6398,    Adjusted R-squared:  0.6326
F-statistic: 88.17 on 75 and 3722 DF,  p-value: < 2.2e-16

> vif(model3_local)
                   GVIF Df GVIF^(1/(2*Df))
HomeTeam       8.513622 34        1.031996
AwayTeam       4.846896 34        1.023482
HS_avg        15.429376  1        3.928024
HC_avg        16.833241  1        4.102833
HP_avg        19.539410  1        4.420341
B365H          4.627317  1        2.151120
B365D          3.607529  1        1.899350
HS_avg:HC_avg 39.402926  1        6.277175
HS_avg:HP_avg 31.402303  1        5.603776
```

Ilustración 6: Resumen del modelo y análisis de colinealidad

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Como se puede observar, el modelo obtenido cuenta con un R-cuadrado de 0'64 y un R-cuadrado ajustado de 0'63, unos datos muy similares al modelo final obtenido para la otra variable. Este es un mejor resultado para el conjunto de datos de entrenamiento que el obtenido para el segundo modelo, indicándonos que algo de mejoría se ha obtenido mediante la interacción de términos. Aun así, el principal beneficio se aprecia a la hora de comprobar el rendimiento con los datos de prueba, el cual se estudiará a continuación.

Resultados y análisis

El análisis de los resultados obtenidos para el equipo local y para el equipo visitante se realizará de manera simultánea. En primer lugar, se va a analiza el error cuadrático medio de ambos conjuntos, esta medida nos va a servir para ver la presencia de sobreajuste y ver además qué se predice mejor, si los goles marcados por el equipo local o los goles marcados por el equipo visitante.

Para los goles marcados por el equipo visitante, se obtuvieron los siguientes resultados:

MSE.lr.test	0.2063386863186 86
MSE.lr.train	0.20903434135861

Para los goles marcados por el equipo local se obtuvieron los siguientes resultados:

MSE.lr.test	0.305686340611172
MSE.lr.train	0.313543301059968

En ambos casos podemos comprobar que la presencia de sobreajuste es nula (en ambos modelos se obtiene un mejor rendimiento para el conjunto de prueba que para el conjunto de entrenamiento), el procedimiento sistemático que se ha seguido para tratar de obtener la mejor capacidad de predicción posible con el menor sobreajuste posible ha dado sus frutos.

Por otra parte, se puede apreciar el error cometido a la hora de predecir los goles que va a marcar el equipo visitante es inferior al error cometido a la hora de predecir los goles que va a marcar el equipo local. Quizá este error cometido se deba a un error personal mío, ya que se perdió capacidad de predicción para ambos conjuntos (de prueba y de entrenamiento) al pasar del primer al segundo modelo. Igual yo consideré como no significativas algunas variables que igual si lo eran.

Para el análisis gráfico, en este caso se realizó el mismo análisis para el conjunto de prueba de ambos modelos. Esto se debe a que, debido a la ausencia de sobreajuste en los modelos definitivos, nos era indiferente realizar el análisis sobre un conjunto de datos u otro. Para futuros algoritmos empleados se estudiarán los resultados obtenidos para los diferentes conjuntos de datos.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

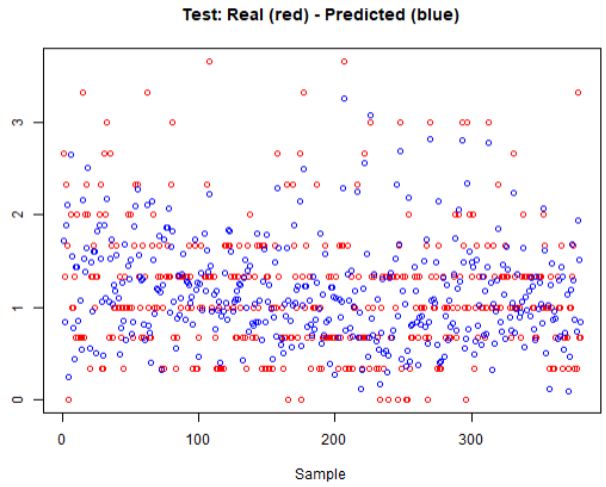


Ilustración 7: Valores reales vs valores predichos modelo visitante

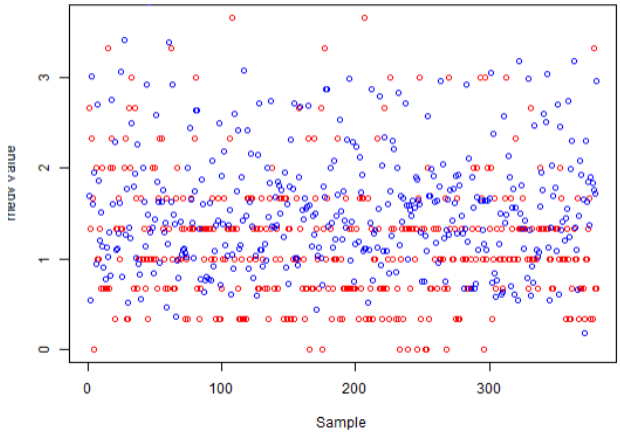


Ilustración 8: Valores reales vs valores predichos modelo local

Si interpretamos las Ilustraciones que muestran la representación gráfica de los valores reales frente a los valores predichos, podemos sacar de ambos casos conclusiones similares. Los valores reales de los datos predichos se encuentran en niveles, como si nos encontrásemos con las predicciones en un árbol de regresión, no obstante, esto se debe a que los goles marcados por un equipo son variables discretas, mientras que las predicciones no lo son. No se han querido redondear la salida al valor discreto más cercano para no alterar el valor obtenido para el error cuadrático medio.

La única diferencia apreciable entre los modelos es que el modelo para el equipo visitante, tiende a no estimar grandes goleadas, ya que vemos una ausencia de puntos en los valores más superiores. Por otra parte, el modelo para predecir los goles locales tiende a estimar siempre una cuantía de goles considerable, prediciendo resultados más abultados de lo que terminan siendo.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

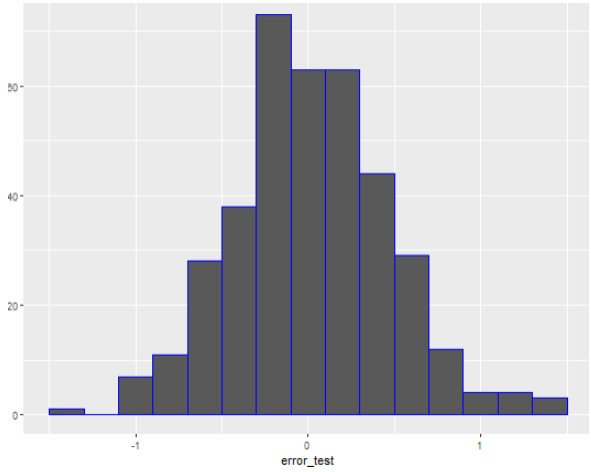


Ilustración 9: Histograma para la variable visitante

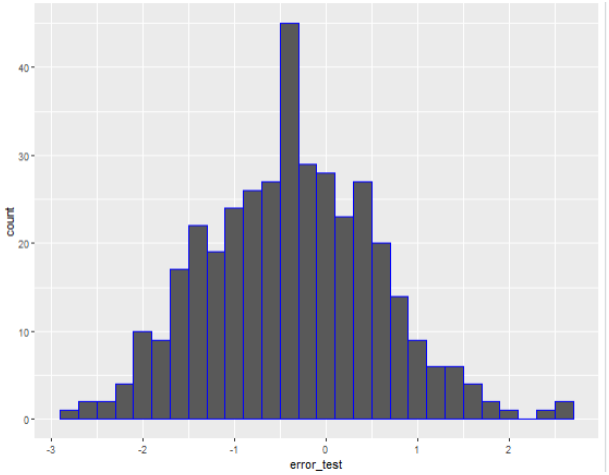


Ilustración 10: Histograma para la variable local

Como se puede observar en el histograma, para ambos casos, los errores están centrados en el cero, se podría afirmar que la distribución de los errores sigue una distribución normal. Pese a que haya algunos casos donde se cometen errores aislados, no se suelen cometer errores superiores a 1 unidad. Tal y como se comentaba en el análisis realizado anteriormente, para los goles visitantes vemos una ligera asimetría a la derecha, tendiendo a infraestimar los goles que van a ser marcados. Se aprecia el efecto contrario para los goles marcados por los equipos locales.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

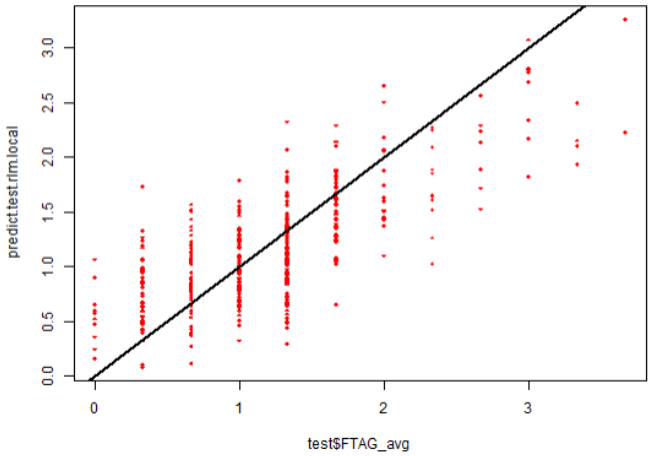


Ilustración 11: Correlación entre la variable real y la variable predicha modelo visitante

En las ilustraciones que muestran la relación entre la variable real y la predicha podemos comprobar que la relación de linealidad entre ambas variables está presente. No obstante, las desviaciones son considerable, siendo los mayores errores cometidos en la zona central de los datos superiores a 1 unidad. Adicionalmente, en esta representación se observan claramente las carencias que se mostraban en los anteriores modelos, mostrándose que para el modelo de los goles anotados por el equipo visitante siempre se tiene a infravalorar el número de goles cuando este es superior a 2 unidades.

La aplicación de este modelo nos ha servido para familiarizarnos con el conjunto de datos, ya que hemos podido seleccionar nosotros de primera mano las variables significativas de ambos modelos. Además, hemos conseguido dos modelos sin sobreajuste que tenían sus respectivas carencias a la hora de predecir unos datos que adopten un valor u otro. Todavía hay posibilidad de mejorar los modelos en cuanto a su precisión, sin sacrificar excesivamente su varianza.

Lasso-Ridge Mixta

Una vez desarrollado el primer modelo, pese a la ausencia de sobreajuste en el mismo, he decidido presentar únicamente el modelo Lasso-Ridge para la variable que describe a los goles locales. Tenía la mosca detrás de la oreja debido a la gran diferencia en el rendimiento obtenida en el apartado anterior. Por tanto, el uso de las técnicas de Lasso y Ridge iban a aportarme un método sistemático donde no puede haber ningún error humano a la hora de determinar qué variables debo considerar como significativas y cuáles no.

En la herramienta de R desarrollé los algoritmos para ambos modelos, no obstante, solo se obtuvieron resultados relevantes para la variable por la cual empleó este algoritmo en primera instancia, es por eso que solo se va a mostrar la parte del análisis que interesa.

En primer lugar, cabe resaltar que he decidido aplicar solamente el modelo de regularización mixto en lugar de aplicar Lasso y Ridge de manera individual porque este modelo ya contempla ambos casos en uno solo modelo. En los párrafos siguientes se explica el funcionamiento del mismo:

Esta forma de regularizar surgió porque en el modelo Lasso la selección de variables puede llegar a ser muy dependiente del conjunto de datos de entrenamiento, lo que provoca inestabilidad en el modelo. Para ello se juntaron las formas de regularización de Lasso y de Ridge, obtenido la fórmula presentada al final del párrafo. En dicha fórmula nos seguimos basando en el mismo principio donde modificamos

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

levemente la fórmula del error cometido (la cual es empleada para obtener los coeficientes de nuestro modelo). En este caso, α es un parámetro de mezcla que puede oscilar entre cero y uno, cuando vale 0 el tipo de regularización es Ridge, y cuando vale 1 el tipo de regularización es Lasso.

$$\sum_{j=1}^{N}\left(y_j-\sum_{i=0}^{p}\beta_i x_{i,j}\right)^2+\lambda_1(1-\alpha)\sum_{i=1}^{p}{\beta_i}^2+\lambda_2\alpha\sum_{i=1}^{p}|\beta_i|$$

Ilustración 12: Fórmula de regularización tipo Mixto

Análisis con R

A la hora realizar el análisis mixto, para poder apreciar la mejor combinación entre los parámetros α y λ, y así poder obtener el mejor modelo, he creado un vector con diferentes valores de α que van desde 0 hasta uno con una diferencia de 0'1 entre cada elemento. Para los valores de λ he considerado que sus valores oscilan desde 0'01 y 2000.

En la Ilustración inferior se puede apreciar que a medida que aumenta el parámetro de regularización, el error cometido aumenta hasta estancarse. Por tanto, la necesidad de reducir la influencia de las variables explicativas es prácticamente nula. Por otra parte, a la hora de diferenciar si es más beneficioso emplear la técnica de Lasso o la técnica de Ridge de manera individual, nos es indiferente ya que, para el valor escogido para el parámetro de regularización, el error cometido es prácticamente idéntico para ambos casos.

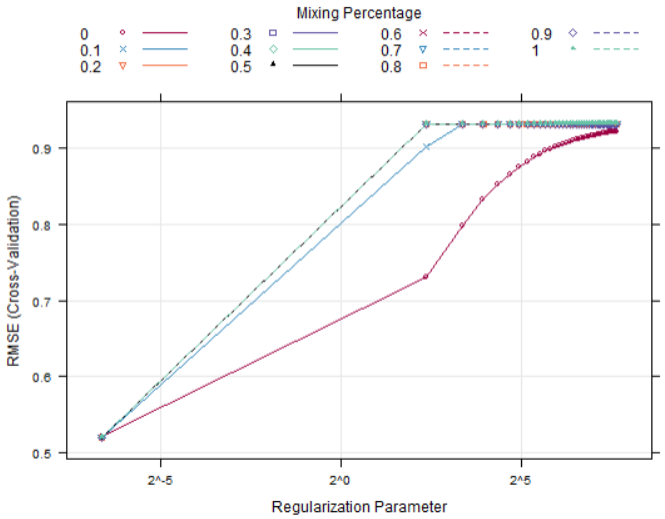


Ilustración 13: Rendimiento del modelo en función de la combinación de parámetros

En el punto óptimo donde se reduce en mayor medida el error cuadrático cometido medio para el conjunto de datos de entrenamiento, se han obtenido los siguientes valores para los parámetros estudiados.

Resultados y análisis

En primer lugar, se va a analiza el error cuadrático medio, esta medida nos va a servir para ver la presencia de sobreajuste, además podremos ver la capacidad de predicción del modelo en comparación con los resultados obtenidos anteriormente.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Para los goles marcados por el equipo local, se obtuvieron los siguientes resultados:

MSE.lr.test	0.258090011169964
MSE.lr.train	0.262832482661001

En el caso estudiado podemos comprobar que la presencia de sobreajuste es nula (se obtiene un mejor rendimiento para el conjunto de prueba que para el conjunto de entrenamiento), el procedimiento seguido por el algoritmo para seleccionar los parámetros ha dado un mejor resultado que el procedimiento personal de selección de variables y de interacciones.

Para el análisis gráfico, como en el caso anterior se va a realizar exclusivamente para el conjunto de prueba. Esto se debe a que, debido a la ausencia de sobreajuste en los modelos definitivos, nos era indiferente realizar el análisis sobre un conjunto de datos u otro.

En este caso, a diferencia de los estudios gráficos realizados anteriormente, vemos unas relaciones considerablemente más consideradas. En las 3 representaciones podemos apreciar que el modelo no suele sobreestimar ni infraestimar los goles marcados (Histograma centrado en el cero, sin asimetrías, y los errores oscilan entre +1 y -1 unidad. Misma cantidad de casos aproximadamente por encima de la línea que resultaría en la predicción perfecta). Además, al igual que en el caso anterior, vemos una relación directamente proporcional entre los resultados reales y experimentales.

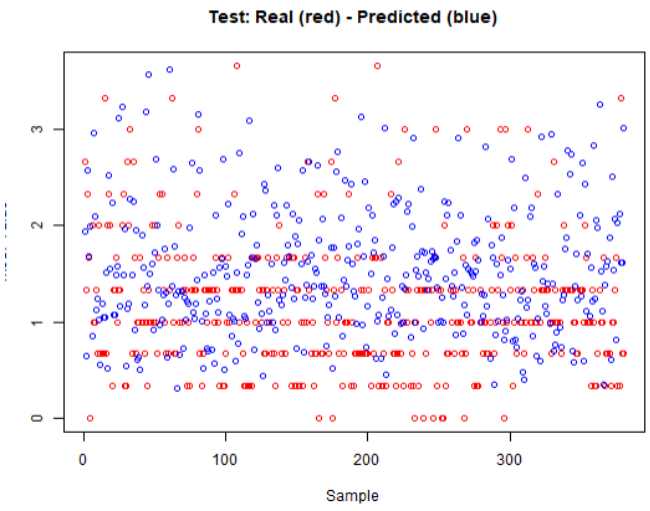


Ilustración 14: Valores reales vs valores predichos modelo local

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

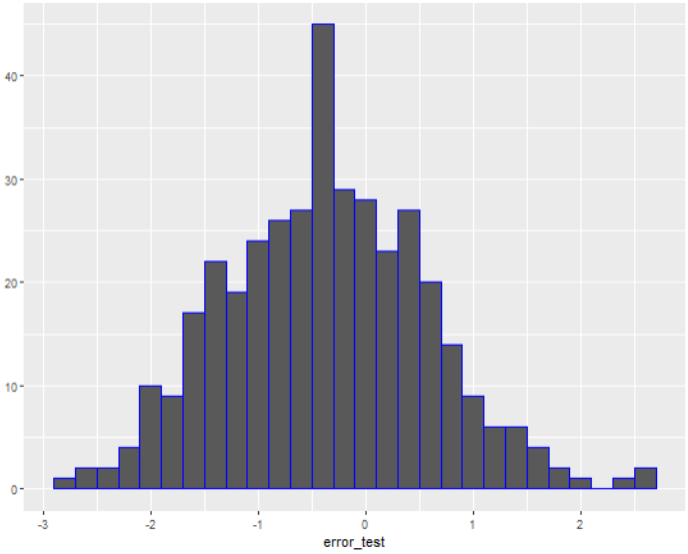


Ilustración 15: Histograma de residuos para el modelo local

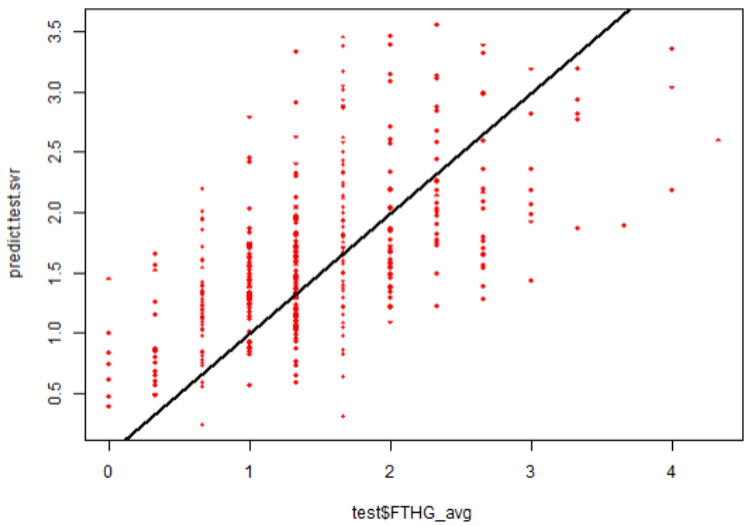


Ilustración 16: Correlación entre la variable real y la variable predicha para el modelo local

Mediante el uso de la regularización se ha obtenido un modelo considerablemente más fiable que ha provocado que mejorase nuestro modelo en dos aspectos fundamentales. En primer lugar, se ha conseguido mejorar la capacidad de predicción manteniendo el sobreajuste a raya. En segundo lugar, hemos conseguido que las predicciones realizadas por regla general no tiendan a sobrestimar los goles marcados, desembocando en resultados más abultados.

Red neuronal (el perceptrón multicapa)

Tras realizar los dos primeros modelos, he decidido emplear el algoritmo del perceptrón multicapa porque es el único algoritmo que me permite crear un único modelo que me permite predecir las dos variables al mismo tiempo. Esto supondría un ahorro de tiempo a la hora de su implementación, ya que los datos no deben entrar en dos modelos distintos para obtener un resultado separado. Además,

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

al ser un único modelo, debido a la disposición de los pesos de la red neuronal, habrá una mayor relación entre las dos variables a predecir.

Se forma cuando conectamos varias neuronas perceptrón en diferentes capas formando una red. Las capas pueden ser de entrada (donde se introduce el estímulo), capa escondida y capa de salida (la salida de estas neuronas es la salida de la red). El proceso de aprendizaje de la red consiste de dos etapas, la etapa de entrenamiento (donde adaptamos los pesos sinápticos mediante estímulos) y la etapa de validación.

El algoritmo que aplica esta red neuronal para ajustar sus pesos es el de back-propagation. Este algoritmo se caracteriza en que la corrección del peso es proporcional al gradiente instantáneo. Dicho gradiente local se calcula de manera diferente en función de si nos encontramos en una neurona de la capa de salida o en una neurona de la capa intermedia. Para obtener estos pesos debemos que van a determinar la salida de la red debemos considerar factores como el número de neuronas de la red, el factor de aprendizaje, la función de activación (la cual debe ser continua y derivable), los estímulos empleados y el número de épocas.

Una época ocurre cuando se presenta un conjunto de datos de entrenamiento entero a la red. Este proceso se repite hasta que el error converja al valor mínimo. Adicionalmente, hay que considerar la capacidad de generalización del modelo para evitar el sobreajuste y mantener su capacidad de generalización. Los dos principales métodos de aprendizaje son:

1) Patrón: Los pesos se ajustan cada vez que se presenta un caso nuevo.
2) Lote: Los pesos se ajustan una vez han pasado todos los casos de una época.

Hay varias técnicas para determinar cuándo parar de entrenar a la red neuronal. Cuando el error por ciclo deje de variar o varíe mínimamente, cuando el error llegue a un umbral, cuando dejen de variar los pesos sinápticos o cuando generalice adecuadamente mi modelo.

A la hora de inicializar los pesos de las neuronas, es importante que haya aleatoriedad para que la red no sature rápidamente. Para ello es recomendable normalizar, ya que nos aseguramos que las neuronas trabajen en una zona lineal adecuada, provocando un aprendizaje más homogéneo y evitamos problemas con los rangos de los atributos.

Análisis con RStudio

Como se puede apreciar en la Ilustración que se muestra a continuación, aproximadamente en 50 iteraciones se estabiliza el error mínimo cometido. De esta manera, voy a seleccionar ese valor como el máximo a estudiar porque si escojo un valor mayor, puede ser que pierda capacidad de generalización y aumente el sobreajuste. El modelo obtenido como máximo tendrá 50 épocas, no obstante, puede que su valor óptimo sea menor.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

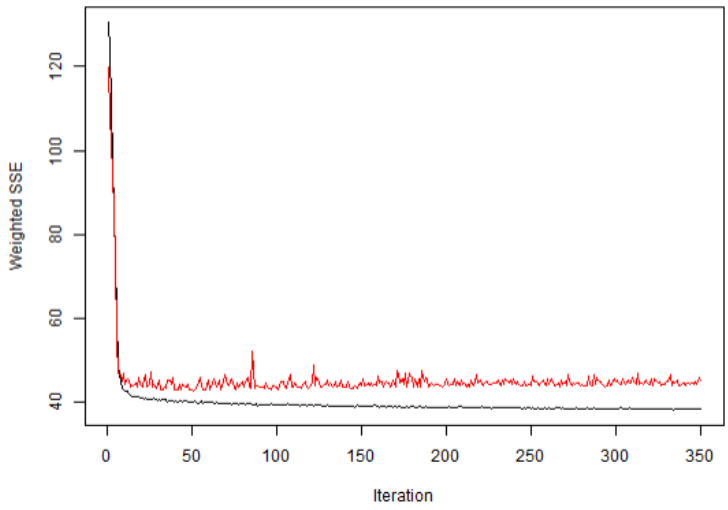


Ilustración 17: Número óptimo de iteraciones

Como comentario final, las medidas obtenidas para la ilustración 1 son para una red neuronal con 1 capa intermedias de 3 neuronas. Además, tendremos 18 neuronas en la capa de entrada, una para cada variable (no he considerado las variables de equipo local ni equipo visitante porque el gran número de variables dummies para dichas variables formaban una red neuronal enorme), y 2 neuronas para la capa de salida, cada se corresponderá con su salida predicha. A la hora de elegir el número de neuronas en las capas intermedias, me fijé en el error cuadrático medio a la hora de predecir con el conjunto de datos de entrenamiento y con el conjunto de datos de test. Para ellos probé diferentes combinaciones, y el aquellas redes neuronales con un mayor número de neuronas en las capas intermedias tenían una gran capacidad de predicción para el conjunto de datos de entrenamiento, en comparación con el rendimiento de los modelos ya estudiados. No obstante, estas redes neuronales tenían una pésima capacidad de predicción para el conjunto de test, presentando un gran sobreajuste, es por eso que he decidido emplear una red neuronal bastante simple. Adicionalmente, si me decantaba por una red neuronal excesivamente simplificada en las capas intermedias, también tenía un peor rendimiento el modelo.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

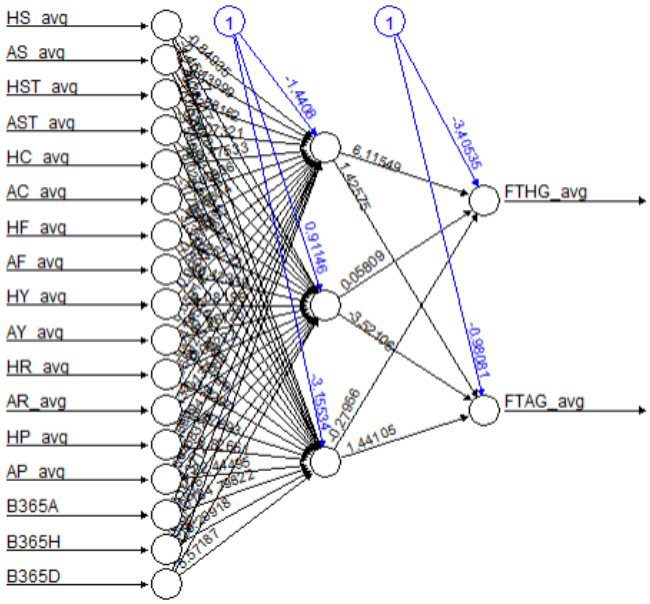


Ilustración 18: Red Neuronal Obtenida

Para la red mostrada en la ilustración anterior, se obtuvo que el número óptimo de iteraciones donde se obtuvo el mínimo error fueron 7, además, la función de activación óptima para todas las neuronas fue la radial.

```
> nn$weights[best_rep]
[[1]]
[[1]][[1]]
              [,1]        [,2]        [,3]
 [1,] -1.440795832  0.91146110   -3.155337
 [2,] -0.849352124 -0.45816785  -12.340100
 [3,]  0.139987848  0.38217403  -18.052632
 [4,]  1.681620600  0.87783935   -3.596855
 [5,]  0.073212070 -3.13702211   18.027401
 [6,] -0.175331344 -0.55956756   -1.798712
 [7,]  0.009356890  0.48621856    3.302467
 [8,] -0.066038063 -0.25272176   -5.184002
 [9,] -0.072891925 -0.42339257   -1.436521
[10,]  0.095508718  0.08199497    3.801086
[11,]  0.057166266 -0.26724939   -6.017987
[12,]  0.041446067 -0.09221952   -3.329695
[13,] -0.066921439 -0.13088449    6.676933
[14,]  0.843300381  0.39427722   -6.826606
[15,]  0.145931486 -2.68325916   22.444952
[16,]  0.005620561 -1.43712350 -164.798222
[17,] -0.531219129 -1.76628423   -9.299185
[18,]  0.990981098  2.44292246    5.571865

[[1]][[2]]
            [,1]       [,2]
[1,] -3.40534872 -0.9808124
[2,]  6.11548953  1.4257492
[3,]  0.05809395 -3.5210559
[4,] -0.27955643  1.4410497

>
```

Ilustración 19: Pesos de las neuronas

A la hora de interpretar los pesos, sabiendo que todos los valores están en la misma escala, podemos afirmar que aquellas neuronas de entrada con mayores pesos sean un indicador de que variables son

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

más significativas. No obstante, como hay un gran número de interconexiones, y al haber dos salidas, no se pueden sacar conclusiones claras.

Resultados y análisis

El análisis de los resultados obtenidos para el equipo local y para el equipo visitante se realizará de manera simultánea. En primer lugar, se va a analiza el error cuadrático medio de ambos conjuntos, esta medida nos va a servir para ver la presencia de sobreajuste y ver además qué se predice mejor, si los goles marcados por el equipo local o los goles marcados por el equipo visitante. Por otra parte, estas medidas nos van a permitir comparar el rendimiento de este modelo con los desarrollados anteriormente.

Para los goles marcados por el equipo visitante, se obtuvieron los siguientes resultados:

MSE conjunto de entrenamiento: 0.250

MSE conjunto de test: 0.298

Para los goles marcados por el equipo local se obtuvieron los siguientes resultados:

MSE conjunto de entrenamiento: 0.285

MSE conjunto de test: 0.315

En ambos casos podemos comprobar que la presencia de sobreajuste es leve (en ambos modelos se obtiene un rendimiento similar para el conjunto de prueba y para el conjunto de entrenamiento), a la hora de seleccionar la neurona y su tamaño lo he intentado seleccionar el mejor número de capas intermedias el número de combinaciones de las mismas de manera arbitraria, seguramente si hubiese empleado algún tipo de recurso recursivo el resultado habría sido mejor.

Para el análisis gráfico, en este caso se realizó el mismo análisis para el conjunto de prueba de ambos modelos. Esto se debe a que, debido a la ausencia de sobreajuste en los modelos definitivos, nos era indiferente realizar el análisis sobre un conjunto de datos u otro. Para futuros algoritmos empleados se estudiarán los resultados obtenidos para los diferentes conjuntos de datos.

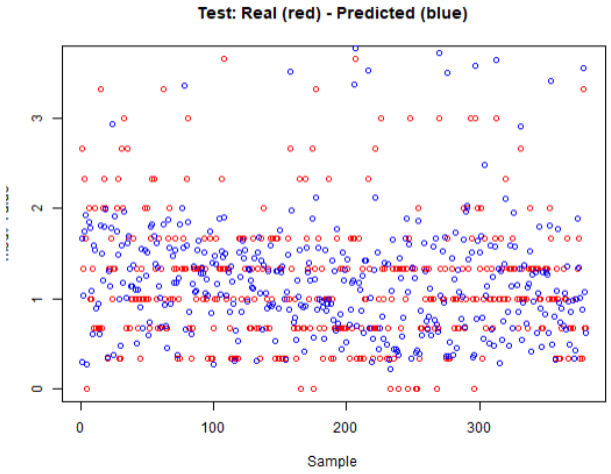


Ilustración 20: Correlación entre la variable real y la predicha modelo visitante

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

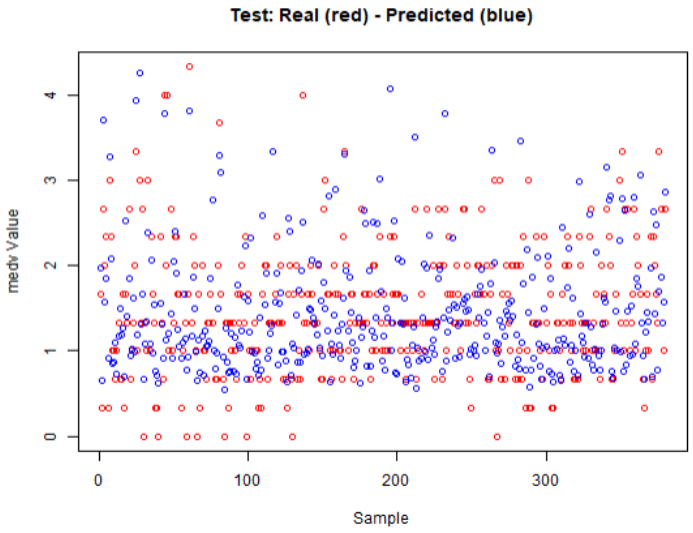


Ilustración 21: correlación entre la variable real y la predicha modelo local

En cuanto a la representación que representa gráficamente el resultado real y el predicho, vemos dos limitaciones, una en cada variable a predecir. Para la variable visitante, vemos que hay un gran número de casos intermedios que no han sido cubiertos, es como si la región entre dos y 3 unidades no hubiese casos prácticamente, por regla general, se ha despreocupado de las unidades superiores. Por otra parte, para la variable local, nos hemos despreocupado de los goles inferiores. La naturaleza de estos errores podría llevarme a pensar que he cometido un error a la hora de desnormalizar las medidas obtenidas, no obstante, en ambas predicciones podemos observar casos aislados que descartan esta hipótesis.

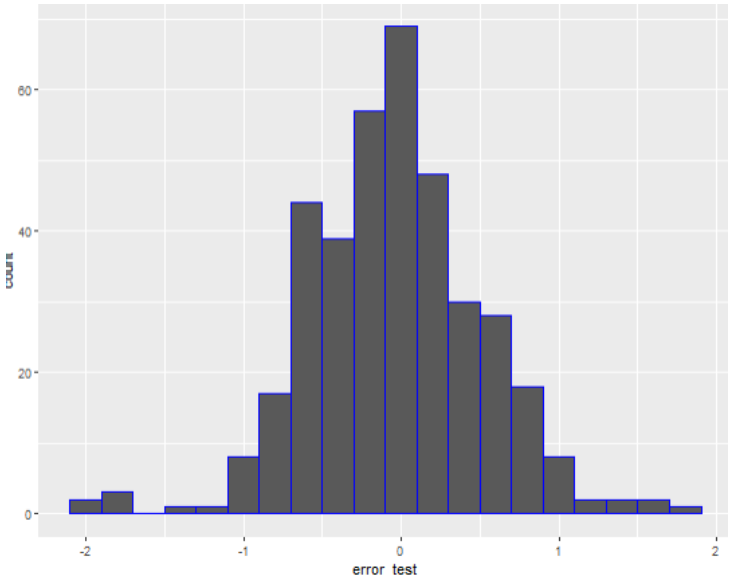


Ilustración 22: Histograma de residuos variable visitante

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

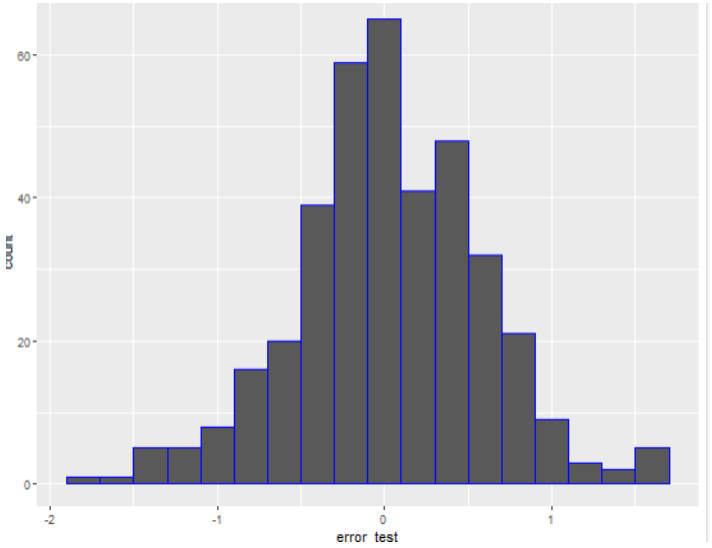


Ilustración 23: Histograma de residuos variable local

Los histogramas muestran las asimetrías correspondientes que se relacionan con los rendimientos de los anteriores modelos. El comportamiento de estas asimetrías ya se discutió en el primer modelo estudiado. Como apunte final, los errores cometidos están centrados en el cero.

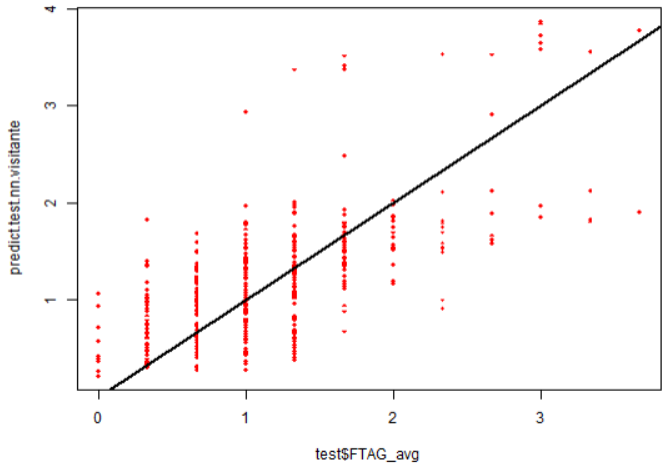


Ilustración 24: Correlación entre medida real y predicha modelo visitante

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

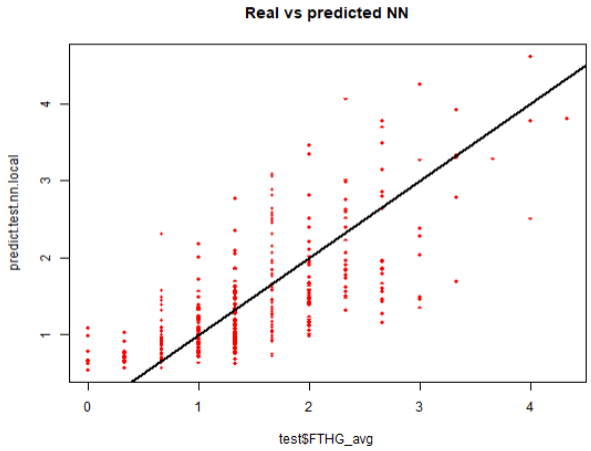


Ilustración 25: Correlación entre medida real y predicha modelo local

En las Ilustraciones superiores vemos que para ambos se cometen errores considerables independientemente del valor estudiado. Siendo más difícil que en otros modelos apreciar la relación lineal entre el valor real y predicho. Este hecho explica porque este modelo tiene un mayor MSE que los estudiados anteriormente.

Considero que, al intentar tratar de realizar un único modelo para facilitar nuestra vida a la hora de aplicarlo a casos reales, hemos visto que el resultado ha sido el peor hasta ahora. Yo creo que la principal causa del fracaso a la hora de predecir radica en la diferencia de variables explicativas significativas de cada una las salidas. Como se ha podido comprobar en el modelo de regresión, cada variable a predecir era explicada por variables muy distintas, pudiendo ser esta la principal causa de confusión en la distribución de los pesos de una red neuronal común.

Máquinas de Soporte Virtuales

En primer lugar, se va a explicar de manera introductoria el funcionamiento de las máquinas de soporte para modelos de regresión.

Las máquinas de soporte virtuales con regresión mantienen la idea fundamental de las de clasificación. Tratan de minimizar el error, individualizando el hiperplano que maximiza el margen (El ancho del hiperplano que puede ser aumentado hasta los vectores de soporte), teniendo en cuenta que parte del error es tolerado.

En SVR se basa en la función del error cometido en valor absoluto, obviando los errores cometidos por debajo de un umbral. Solo contribuyen a la función de coste aquellos puntos que se encuentran por encima del umbral de error cometido.

La principal característica de las máquinas de soporte virtuales es que tan solo influyen en el establecimiento del hiperplano característico los vectores de soporte. Por tanto, para un conjunto de datos tan grande como el que estamos tratando, en un principio es una técnica apropiada ya que se considerarían un gran número de puntos y el modelo establecido tendría una alta capacidad de generalización (el cual es un principal inconveniente a la hora de usar esta técnica). No obstante, a la hora de tratar de implementar el modelo para predecir cada variable por separado me encontré con un inconveniente, el tiempo de procesamiento para la construcción del modelo.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

A la hora de construir el modelo para determinar los parámetros de coste, épsilon y gamma (se explican más adelante), además de determinar los vectores de soporte que me aportan el hiperplano óptimo, dejé el ordenador procesando esta acción toda la noche, sin llegar a converger el problema. Para tratar de proponer alguna alternativa y así presentar algo en el trabajo, en primer lugar, reduje las variables significativas a las obtenidas en el modelo 2 de regresión lineal, de este modo habría menos dimensiones que considerar en la construcción del hiperplano. No obstante, el principal problema se encontraba en la gran cantidad de datos del conjunto de entrenamiento provocando que la elección de los vectores de soporte fuese lenta, por tanto, decidí reducir mi conjunto de entrenamiento a un número reducido de datos.

A la hora de presentar los resultados y el análisis de este modelo, se va a efectuar primero el análisis íntegro de una variable, y a continuación el análisis íntegro de la otra variable.

Análisis con RStudio

Variable visitante

Usando un conjunto de datos de entrenamiento reducido, tal y como se muestra en el código de la figura mostrada a continuación, en la primera parte me centré en la elección de los parámetros del modelo.

```
set.seed(1)
svr_train<-tune("svm",FTAG_avg ~ HomeTeam+AwayTeam+AS_avg+AST_avg+
                  AC_avg+AP_avg+B365A+B365H+B365D,
                data=train[1:200,],
                kernel="radial",
                ranges=list(
                  epsilon=seq(0,1,0.1),
                  cost=c(0.01, 0.1, 1, 5, 10, 20),
                  gamma=c(0.1, 0.5, 1, 2, 5, 10)))
```

Ilustración 26: Construcción del modelo

Los parámetros a determinar eran el parámetro épsilon (el margen de tolerancia), el parámetro de coste (el que controla el equilibrio entre el bias y la varianza), y el parámetro gamma (mide la desviación típica dentro de la función gausiana empleada como Kernel).

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

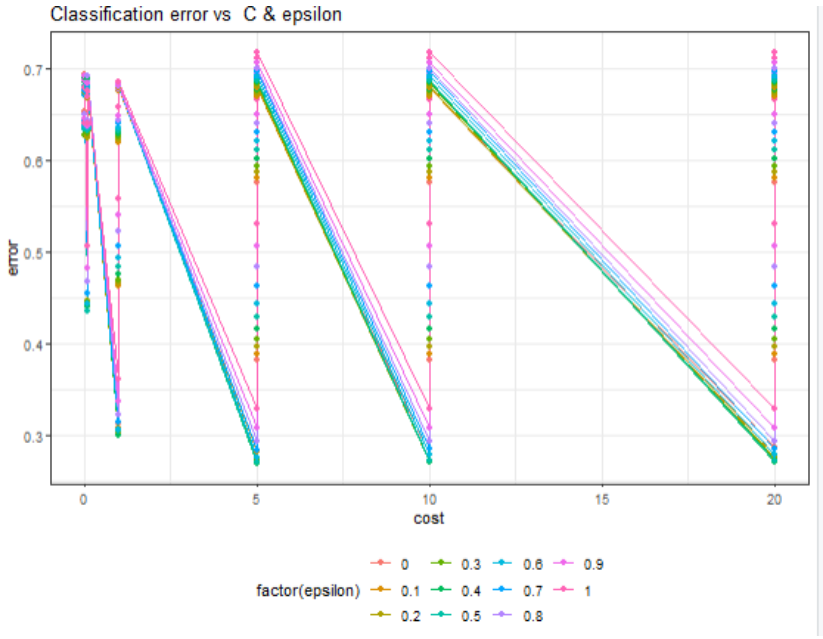


Ilustración 27: Elección de parámetros SVR

Para la elección de la mejor combinación de los parámetros se probaron diferentes combinaciones dentro de los rangos adecuados correspondientes, y el criterio de mejor elección fue aquella combinación que minimizase el error cometido. En este caso se obtuvo que los mejores parámetros fueron:

```
- best parameters:
 epsilon cost gamma
     0.4    5   0.1
```

Ilustración 28: parámetros óptimos

```
> print(W)
     HomeTeamAlaves HomeTeamAlmeria HomeTeamAth.Bilbao HomeTeamAth.Madrid HomeTeamBarcelona HomeTeamBetis HomeTeamCelta HomeTeamCordoba HomeTeamEibar HomeTeamElche HomeTeamEspanol HomeTeamGetafe HomeTeamGirona
[1,]             0       2.154038        -0.1090995        -0.5336081        -2.334356             0             0               0             0             0      -0.6856823       2.899119             0
     HomeTeamGranada HomeTeamHercules HomeTeamHuesca HomeTeamLa.Coruna HomeTeamLas.Palmas HomeTeamLeganes HomeTeamLevante HomeTeamMalaga HomeTeamMallorca HomeTeamOsasuna HomeTeamReal.Madrid HomeTeamSantander
[1,]             0                0              0        -1.432554                 0              0              0               0      -0.2044714     -0.6586739      -1.697375      -0.7362974       3.130529
     HomeTeamSevilla HomeTeamSociedad HomeTeamSp.Gijon HomeTeamTenerife HomeTeamValencia HomeTeamValladolid HomeTeamVallecano HomeTeamVillarreal HomeTeamXerez HomeTeamZaragoza AwayTeamAlmeria AwayTeamAth.Bilbao
[1,]      -1.016591              0       2.350603       -2.062506       -1.288157       4.120142              0       0.4814943     -2.128924      -0.2476283     0.02568002      -4.323234
     AwayTeamAth.Madrid AwayTeamBarcelona AwayTeamBetis AwayTeamCelta AwayTeamCordoba AwayTeamEibar AwayTeamElche AwayTeamEspanol AwayTeamGetafe AwayTeamGirona AwayTeamGranada AwayTeamHercules AwayTeamHuesca
[1,]      -0.4852909       -0.02952087              0              0              0             0             0      -2.724717      2.461344             0              0              0             0
     AwayTeamLa.Coruna AwayTeamLas.Palmas AwayTeamLeganes AwayTeamLevante AwayTeamMalaga AwayTeamMallorca AwayTeamOsasuna AwayTeamReal.Madrid AwayTeamSantander AwayTeamSevilla AwayTeamSociedad AwayTeamSp.Gijon
[1,]      -0.1203727              0              0              0             0       1.36703       0.7914999       -1.15007        1.383368      2.470796      -0.2627207              0      -0.4949705
     AwayTeamTenerife AwayTeamValencia AwayTeamValladolid AwayTeamVallecano AwayTeamVillarreal AwayTeamXerez AwayTeamZaragoza   AS_avg   AST_avg    AC_avg   AP_avg     B365A     B365H     B365D
[1,]       -1.844104        4.712554       -0.3041686              0       -1.004289     -2.247555       0.7787404 5.968325 19.67912 -1.677301 25.20022 -9.132308 13.42147 0.03340928
```

Ilustración 29: Coeficientes de las variables explicativas

Sería precipitado decir que alguna variable es más significativa que otra basándonos exclusivamente en el valor de los pesos. Esto se debe a que desconocemos el rango de cada una de las variables, adicionalmente, a diferencia de los modelos de regresión lineal simple, no contamos una medida como la del p-valor que nos indique la significatividad de la variable.

El error cuadrático medio obtenido para las variables de entrenamiento (considerando el conjunto de entrenamiento el mismo que el empleado en los anteriores algoritmos, no el empleado para entrenar este modelo específico) y de prueba es el siguiente:

MSE.svr.test	0.264767699481332
MSE.svr.train	0.279269469691241

La presencia de sobreajuste en este modelo es nula, esto se debe a que el modelo en ambos casos se está enfrentando en su totalidad o en su mayoría a casos nuevos. Esta situación justifica que a la hora

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

de hacer el análisis gráfico solo se haga con los datos de prueba, porque el resultado obtenido en el otro caso es idéntico.

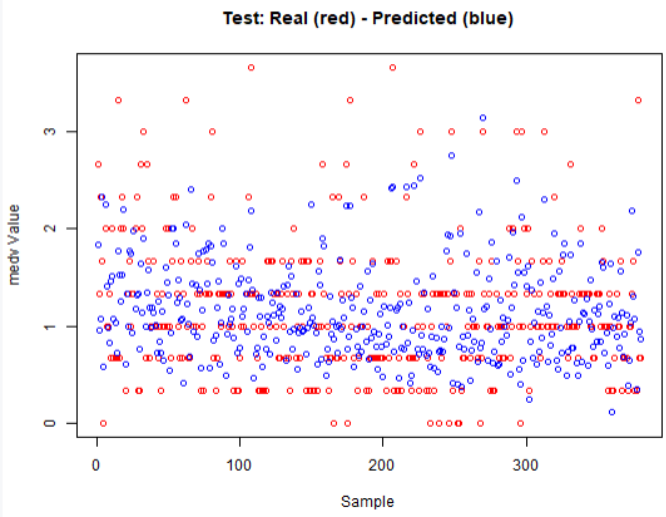


Ilustración 30: correlación entre la variable real y la predicha modelo visitante

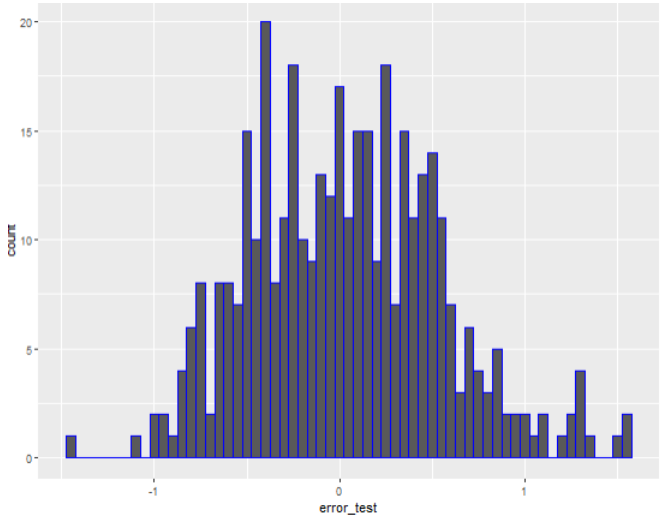


Ilustración 31: Histograma de residuos modelo visitante

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

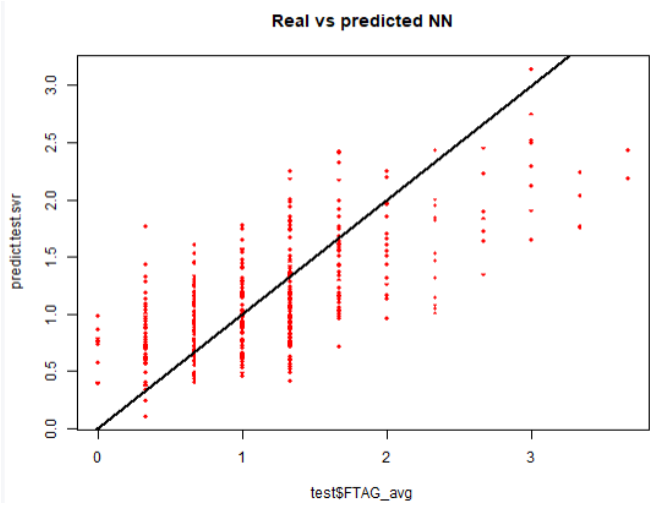


Ilustración 32: Correlación entre medida real y predicha modelo visitante

A la hora de interpretar el análisis gráfico, podemos observar que en su conjunto los errores están centrados en el intervalo de más-menos 1 unidad. No obstante, dada la limitación de tamaño del conjunto de datos de entrenamiento, el modelo clasifica la gran mayoría de los datos donde se encuentra la mayoría de los mismo. No es capaz de clasificar datos aislados o muy diferentes a la media porque nunca se ha enfrentado a estos.

Este hecho explica que el grueso de las predicciones se encuentre en la zona media, además, explica también porque el histograma pese a estar centrado en el cero, comete un gran número de errores al estar centrado en el cero, el histograma de residuos pierde la propiedad de asemejarse a una distribución normal. Por último, este hecho explica porque en la correlación entre la medida real y predicha, la mayoría de los datos predichos encuentren alrededor del valor medio de los mismos.

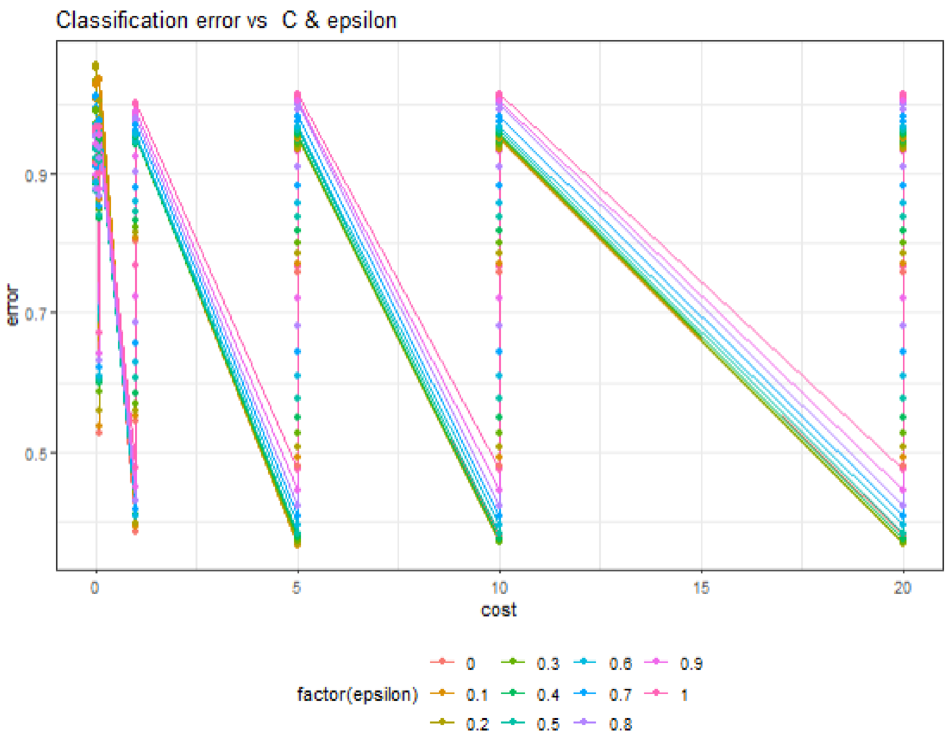
Variable local

Usando un conjunto de datos de entrenamiento reducido, tal y como se muestra en el código de la figura mostrada a continuación, en la primera parte me centré en la elección de los parámetros del modelo.

```
svr_train<-tune("svm",FTHG_avg ~ HomeTeam+AwayTeam+HS_avg+
                  HC_avg+HP_avg+B365H+B365D,
                data=train[1:200,],
                kernel="radial",
                ranges=list(
                  epsilon=seq(0,1,0.1),
                  cost=c(0.01, 0.1, 1, 5, 10, 20),
                  gamma=c(0.1, 0.5, 1, 2, 5, 10)))
```

Los parámetros a determinar eran el parámetro épsilon (el margen de tolerancia), el parámetro de coste (el que controla el equilibrio entre el bias y la varianza), y el parámetro gamma (mide la desviación típica dentro de la función gausiana empleada como Kernel).

María Oliva Calero
Ignacio García Santamaría
3ºB GITT



Para la elección de la mejor combinación de los parámetros se probaron diferentes combinaciones dentro de los rangos adecuados correspondientes, y el criterio de mejor elección fue aquella combinación que minimizase el error cometido. En este caso se obtuvo que los mejores parámetros fueron:

```
- best parameters:
 epsilon cost gamma
     0.1    5   0.1
```

Del mismo modo que en el caso anterior, solo mediante el conocimiento del hiperplano no podemos determinar que coeficientes son los más significativos, porque para eso se requiere el conocimiento del p-valor o de los intervalos de confianza de los mismo.

```
     HomeTeamAlaves HomeTeamAlmeria HomeTeamAth.Bilbao HomeTeamAth.Madrid HomeTeamBarcelona HomeTeamBetis HomeTeamCelta HomeTeamCordoba HomeTeamEibar HomeTeamElche HomeTeamEspanol HomeTeamGetafe HomeTeamGirona
[1,]              0       -4.378304          -2.671009           4.025451          1.982599             0             0               0             0             0       -3.845299      0.7049763              0
     HomeTeamGranada HomeTeamHercules HomeTeamHuesca HomeTeamLa.Coruna HomeTeamLas.Palmas HomeTeamLeganes HomeTeamLevante HomeTeamMalaga HomeTeamMallorca HomeTeamOsasuna HomeTeamReal.Madrid HomeTeamSantander
[1,]               0                0              0          -4.91753                  0               0               0       4.381371         4.191318       -2.347376            4.559697         -1.062507
     HomeTeamSevilla HomeTeamSociedad HomeTeamSp.Gijon HomeTeamTenerife HomeTeamValencia HomeTeamValladolid HomeTeamVallecano HomeTeamVillarreal HomeTeamXerez HomeTeamZaragoza AwayTeamAlmeria AwayTeamAth.Bilbao
[1,]        4.232261                0        -4.110537        -2.425149         1.745632           6.057741                 0           1.813843     -4.936185        -3.000994       -1.600744          -1.063522
     AwayTeamAth.Madrid AwayTeamBarcelona AwayTeamBetis AwayTeamCelta AwayTeamCordoba AwayTeamEibar AwayTeamElche AwayTeamEspanol AwayTeamGetafe AwayTeamGirona AwayTeamGranada AwayTeamHercules AwayTeamHuesca
[1,]           3.708414         0.2239265             0             0               0             0             0      -0.5652234      -1.676152              0               0                0              0
     AwayTeamLa.Coruna AwayTeamLas.Palmas AwayTeamLeganes AwayTeamLevante AwayTeamMalaga AwayTeamMallorca AwayTeamOsasuna AwayTeamReal.Madrid AwayTeamSantander AwayTeamSevilla AwayTeamSociedad AwayTeamSp.Gijon
[1,]         0.1842139                  0               0               0      0.1039833         5.548972        2.583437            1.737377         0.6198579        1.348378                0        -1.478973
     AwayTeamTenerife AwayTeamValencia AwayTeamValladolid AwayTeamVallecano AwayTeamVillarreal AwayTeamXerez AwayTeamZaragoza    HS_avg    HC_avg   HP_avg    B365H    B365D
[1,]         1.537154        0.2592703        -0.07501767                 0         -0.6675073     -4.768637        -5.959207 -2.071039 -1.508871 21.74102 -8.38858 17.52138
```

En este caso, a la hora de medir el rendimiento del modelo tampoco se obtuvo sobreajuste por la misma razón que en el caso anterior. Los resultados obtenidos son peores a la hora de predecir la variable de goles del equipo visitante, este hecho seguramente se deba a la elección del conjunto de datos de entrenamiento. Dado que es una muestra tan pequeña en comparación con el conjunto de prueba real (prácticamente el conjunto del dataset), unos pocos casos que no sean representativos tienen un impacto enorme en la determinación del hiperplano.

MSE.svr.test	0.439661338056147
MSE.svr.train	0.428184006761109

Las representaciones obtenidas para el conjunto de prueba del equipo local se muestran a continuación. Las conclusiones que se extraen son prácticamente idénticas a las obtenidas para la

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

variable anterior, con la única diferencia que se puede inferir que el conjunto de entrenamiento tenía un considerable número de datos de la variable a predecir elevados, ya que se aprecia cierta sobrestimación en todas las gráficas.

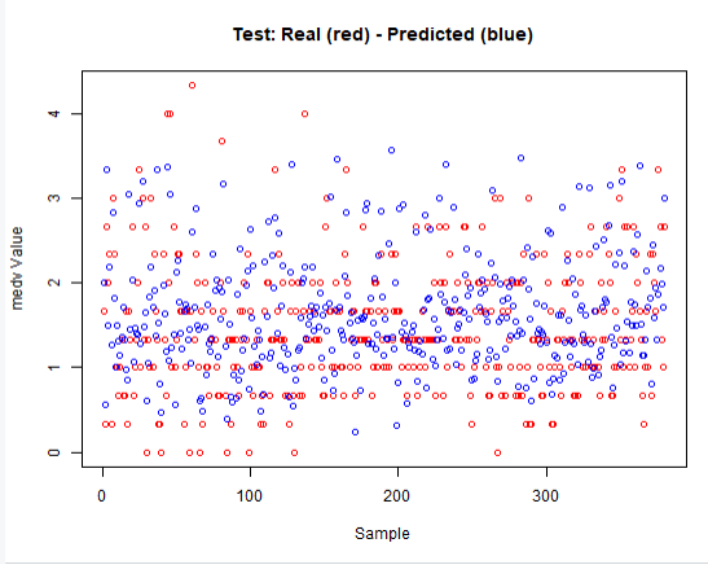


Ilustración 33: correlación entre la variable real y la predicha modelo local

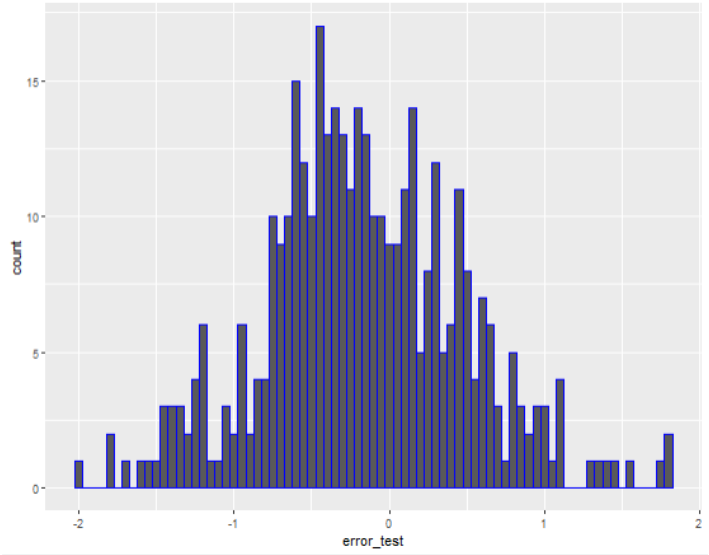


Ilustración 34: Histograma de residuos modelo local

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

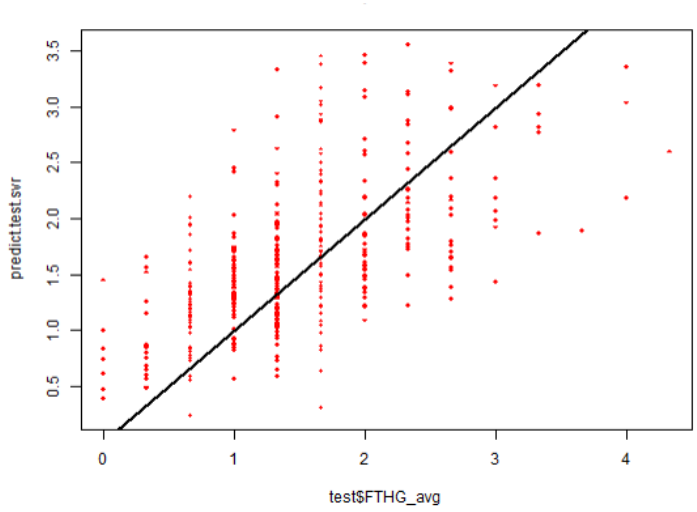


Ilustración 35: Correlación entre medida real y predicha modelo local

Boosting

El principal problema con el que nos hemos encontrado hasta ahora no ha sido el sobreajuste, ha sido la débil capacidad de predicción de los modelos desarrollados hasta ahora. Considero que hay un margen de mejora para estos porque el error cuadrático medio cometido en modelos con gran underfitting, como es el caso de las máquinas de soporte virtuales donde el modelo en sí no es bastante explicativo, se asemeja a resultados obtenidos en otros modelos donde si se ha empleado todo el conjunto de entrenamiento para entrenarlos.

Una razón adicional del uso de esta técnica es que hasta el momento no se han empleado muchos algoritmos que nos den información acerca de las variables más y menos significativas. Debido a la presencia del OOB y a otras razones adicionales, esta técnica aporta información que nos interesa. A continuación, se explica el funcionamiento de la técnica del boosting:

El objetivo de este método es ajustar de manera secuencial varios clasificadores o predictores débiles. En el inicio, partimos de un conjunto de árboles débiles, cada nuevo modelo generado emplea información del anterior. Siempre se emplea el mismo conjunto de entrenamiento.

Estos algoritmos se caracterizan por el gran número de hiperparámetros que utilizan, cuyos valores óptimos se obtienen usando validación cruzada. Los más comunes son:

- El número de clasificadores débiles o de iteraciones, a diferencia que random forest, boosting sí que puede sufrir overfitting si este valor es muy alto.
- Tasa de aprendizaje (λ): Controla como aprenden los modelos, cuanto mayor sea su valor, más árboles serán requeridos y mayor es el riesgo de overfitting.
- El número de divisiones presente en cada árbol.

Adaboost

En este tipo de ensemble los pesos de las observaciones van siendo actualizadas, dándole más peso a la información de los casos mal predichos. Por tanto, el objetivo es solventar los errores producidos. Si un clasificador débil tiene mucho peso es que tiene mucha influencia en el modelo ensemble.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Para poder determinar los valores de los hiperparámetros mencionados anteriormente, ya que en este caso no contamos como en ensembles con el error OOB, voy a tratar de minimizar la tasa de error cometida en el conjunto de entrenamiento realizando validación cruzada.

De manera similar al funcionamiento en el caso anterior, en primer lugar, voy a desarrollar el análisis de una de las variables, explicando más en profundidad el análisis realizado, y en la otra voy a pararme en las cuestiones de interés con el fin de no repetirme.

Análisis con RStudio

Variable visitante

Tasa de aprendizaje

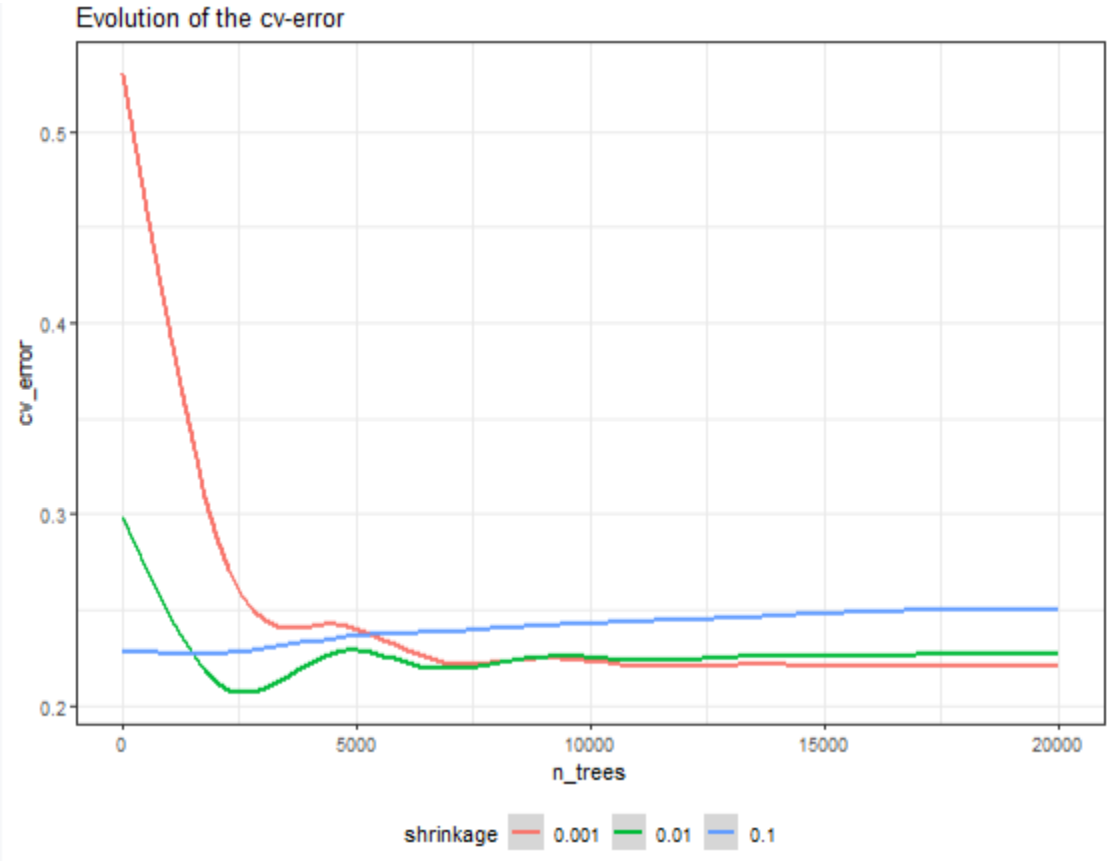


Ilustración 36: Valor óptimo para la tasa de aprendizaje

Esta variable determina la capacidad de aprendizaje del modelo, a medida que este valor es más pequeño hay una mayor probabilidad que mi modelo tenga sobreajuste, por lo tanto, me he decantado por que mi valor sea 0,01 para este hiperparámetro. De este modo obtengo error en validación cruzada considerablemente pequeño, además, siempre voy a tratar de emplear el menor número de árboles posible para reducir el overfitting, por tanto, creo que es el valor que mejor se ajusta.

Complejidad del árbol:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Mediante este parámetro determinamos la complejidad de los árboles usados, cuanto más complejos sean, peor será su capacidad de generalizar, es por eso por lo que me he decantado porque el árbol tenga 4 nodos. De esta manera, me aseguro que el error cometido sea pequeño para el número de árboles que espero (menos que 1000). Dicha medida será determinada tras haber establecido el resto de hiperparámetros.

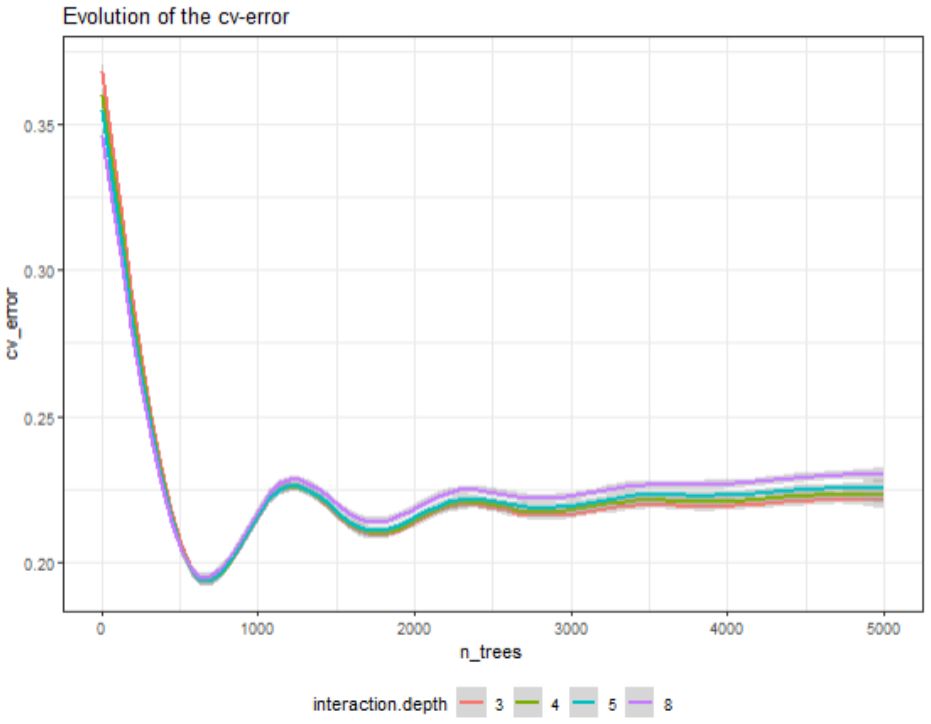


Ilustración 37: Valor óptimo para la profundidad de las interacciones

Número mínimo de observaciones:

Se busca establecer el número de observaciones mínimo que debe haber en cada nodo terminal. En este caso he obtenido que el número óptimo son 10, ya que independientemente del número de árboles que emplee, la comprobación del error mediante validación cruzada será este valor. Adicionalmente, nos aseguramos una buena capacidad de generalización al establecer el valor máximo de las opciones planteadas.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

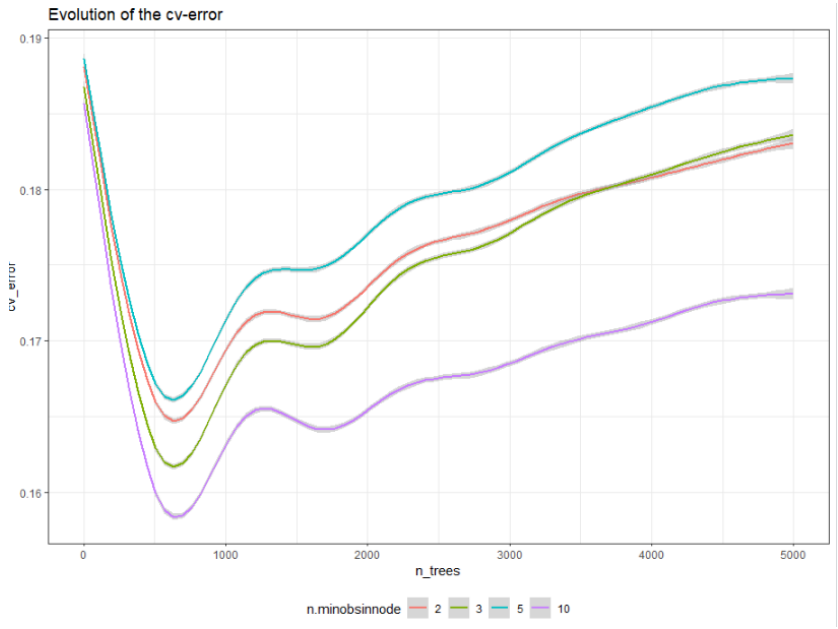
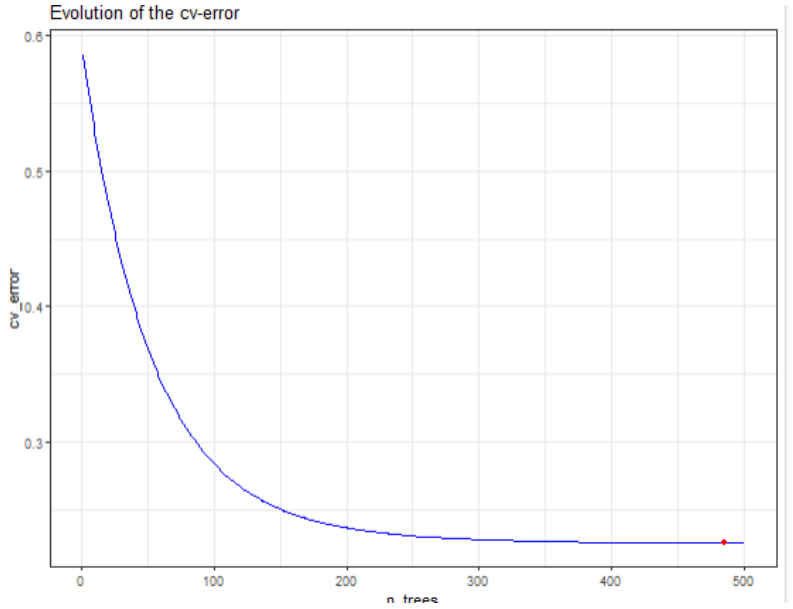


Ilustración 38: Número mínimo de observaciones

Número de árboles:




```
> error[which.min(error$cv_error),]
     cv_error n_trees
485 0.2256741     485
```

Ilustración 39: Número de árboles óptimo

Una vez determinados todos los hiperparámetros, debemos determinar el número de árboles que vamos a emplear para determinar la salida del modelo. Se ha obtenido que donde se comete el mínimo error de validación cruzada es cuando el número de árboles es igual a 485. Vamos a desarrollar 485 árboles con los hiperparámetros ya establecidos para obtener el mejor modelo posible.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

El modelo desarrollado es considerablemente complejo, ya que emplea tanto técnicas de agregación como técnicas de boosting. Las variables que se han considerado más significativas para este último algoritmo siguen el mismo patrón que el obtenido en todos los modelos anteriores, las variables que son consideradas más significativas no han variado.

	var	rel.inf
AP_avg	AP_avg	55.49296930
AST_avg	AST_avg	27.35872914
AwayTeam	AwayTeam	8.71707882
HomeTeam	HomeTeam	5.25120451
AC_avg	AC_avg	1.22121863
AF_avg	AF_avg	0.43530681
B365A	B365A	0.36442088
AY_avg	AY_avg	0.28896521
HS_avg	HS_avg	0.24405416
AS_avg	AS_avg	0.23917257
B365H	B365H	0.14157402
HC_avg	HC_avg	0.06126770
HST_avg	HST_avg	0.05051365
B365D	B365D	0.04508794
HY_avg	HY_avg	0.03398907
HF_avg	HF_avg	0.02610470
AR_avg	AR_avg	0.01795321
HR_avg	HR_avg	0.01038971
HP_avg	HP_avg	0.00000000

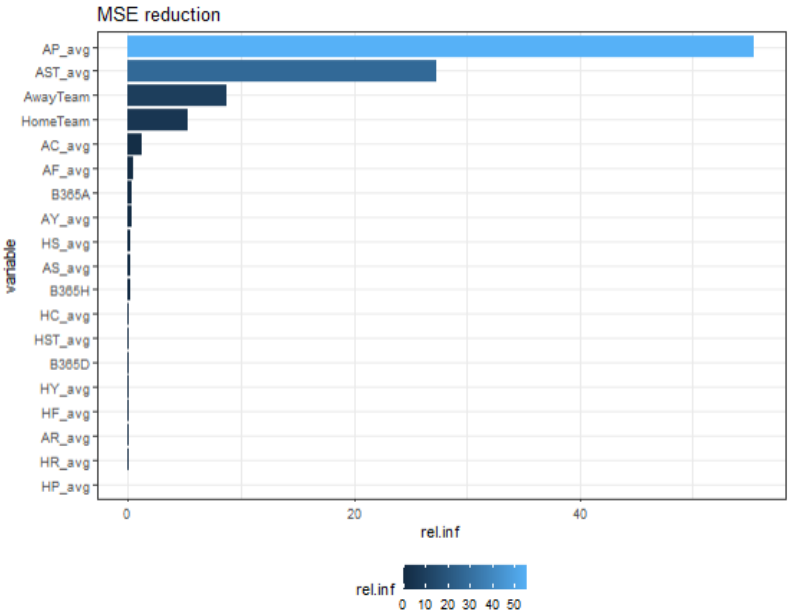


Ilustración 40: Variables más significativas

En la Ilustración que superior se muestra que variables tienen mayor y menor influencia a la hora de reducir el error cuadrático medio considerando el OOB, por tanto, los beneficios obtenidos repercuten tanto el conjunto de entrenamiento como en el conjunto de prueba.

Dentro de las variables más significativas nos encontramos con variables relacionadas con el equipo visitante, tal y como habíamos comentado anteriormente a la hora de establecer el modelo de regresión lineal. No obstante, no todas las variables las variables, justamente aquellas que dan casi por sentado la posibilidad de gol. Si tu obtienes más penaltis "AP_avg" o si obtienes más tiros a puerta "AST_avg", es cuantas más posibilidades tienes de determinar si vas a marcar gol o no. Por otra parte, los equipos que jueguen tienen una gran influencia en el resultado, pero considerablemente menor que el resto de variables comentadas.

A la hora de medir la capacidad de predicción del conjunto de entrenamiento y de prueba, se han obtenido los siguientes resultados:

MSE.rt.test	0.192684660321598
MSE.rt.train	0.190284828068361

Como se puede apreciar, la necesidad de mejorar el rendimiento del modelo es lo que nos ha guiado adecuadamente para escoger esta técnica. Adicionalmente, habiendo tomado las precauciones debidas en la elección de los hiperparámetros, aun habiendo mejorado el rendimiento del modelo, no hay sobreajuste.

A continuación, se va a analizar el rendimiento del modelo gráficamente para el conjunto de datos de entrenamiento y de prueba, de esta manera se va a comprobar la hipótesis que se ha seguido a lo

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

largo del análisis de regresión, la ausencia de sobreajuste implica que los análisis gráficos de ambos conjuntos sean idénticos.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

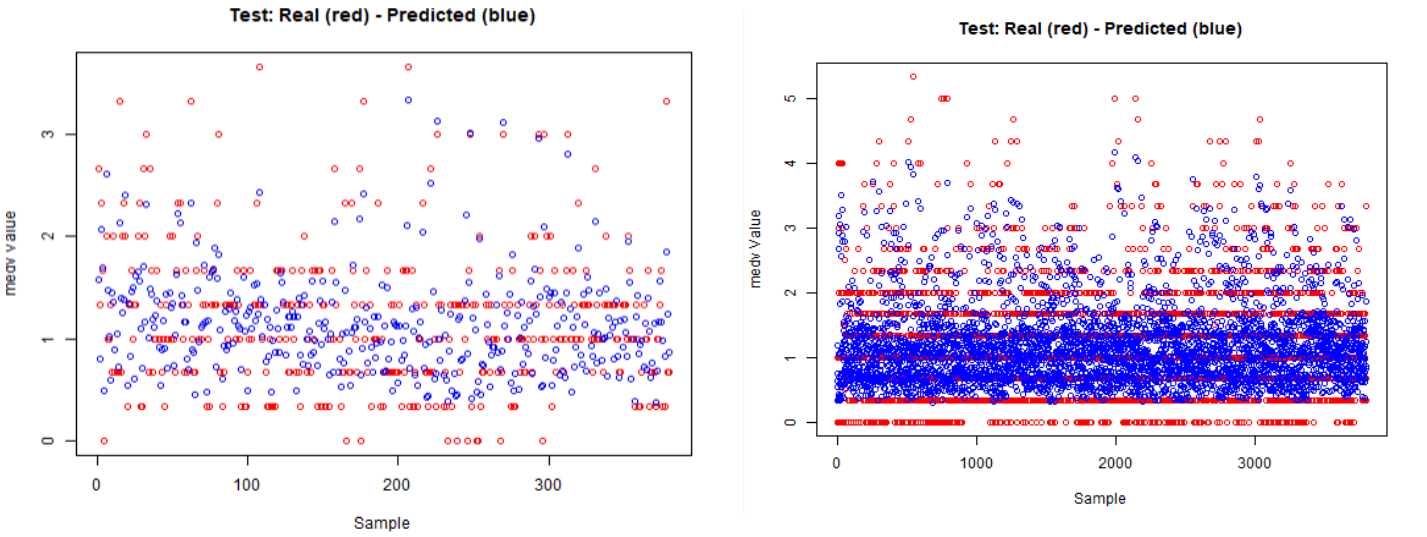


Ilustración 43: correlación entre la variable real y la predicha

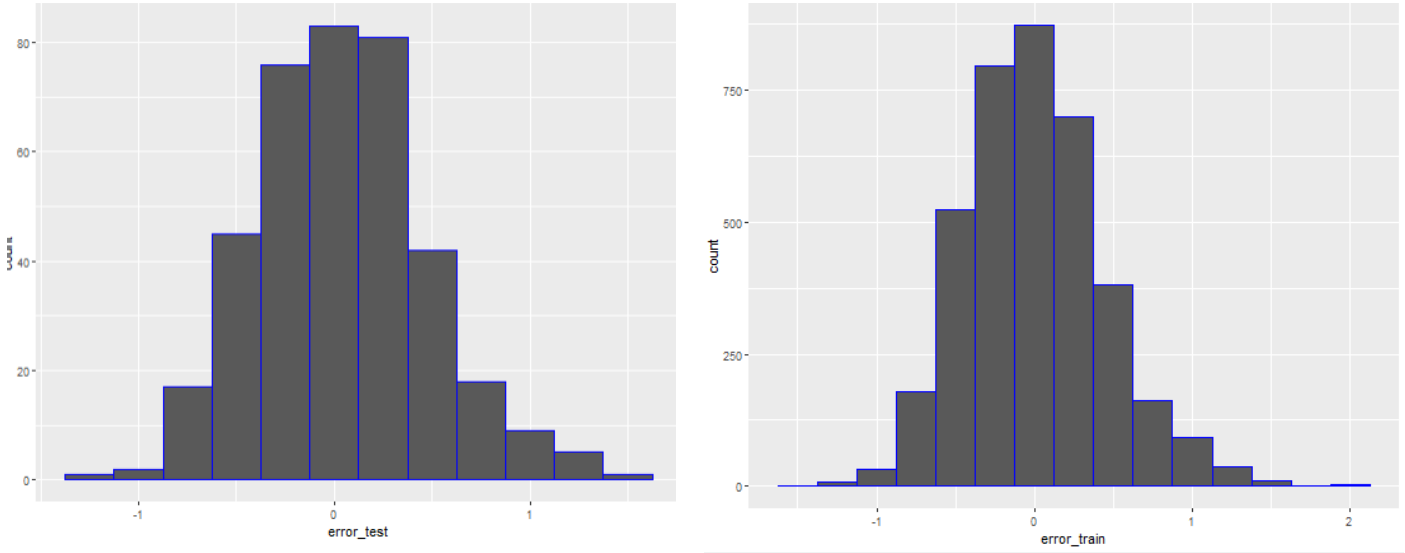


Ilustración 42: Histograma de residuos

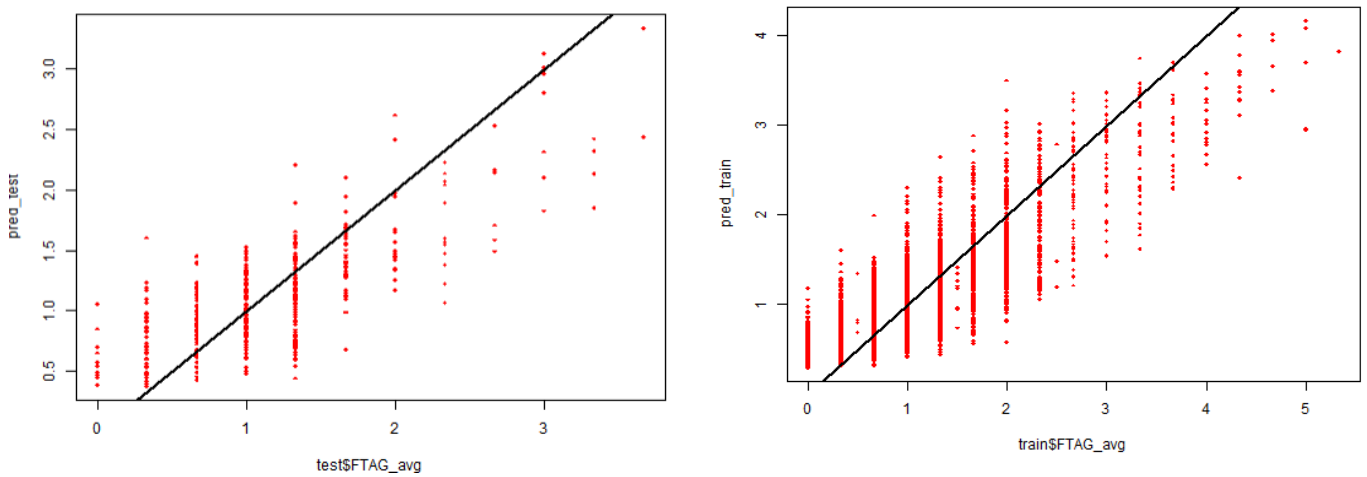


Ilustración 41: Correlación entre medida real y predicha

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

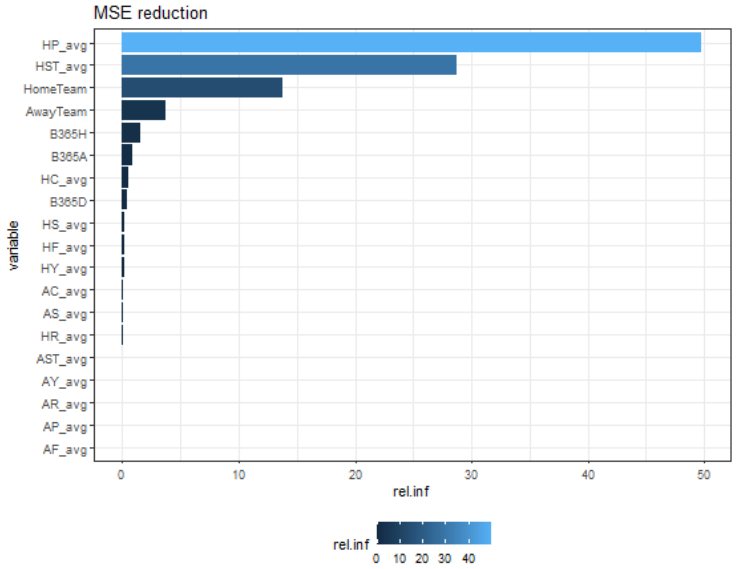
Como se puede comprobar, para el análisis gráfico de los residuos, en lineales generales son aceptables. Hemos obtenido un histograma de residuos prácticamente sin asimetrías, salvo los datos con bajo número de goles, los cuales se suelen sobreestimar, al igual que los datos un alto número de goles que suelen infraestimar, los modelos predicen adecuadamente. Estos son problemas que ya han sido analizados en otros modelos. No obstante, con el análisis gráfico en este caso se ha querido demostrar que, en ausencia de sobreajuste, se muestra la misma imagen por ambos conjuntos, solo que como en el conjunto de entrenamiento hay un mayor número de casos, se aprecia más claramente las fortalezas y debilidades, pero las conclusiones que se extraen son las mismas.

Variable local

A la hora de predecir la variable visitante se realizó el mismo procedimiento, primero determiné los hiperparámetros óptimos, luego se construyó un modelo con dichos hiperparámetros, y al final se midió su capacidad de predicción mediante el error cuadrático medio para ambos conjuntos de datos. Dentro del modelo se consideraron las siguientes variables como significativas.

Como se puede apreciar, a la hora de considerar las variables significativas, se sigue conservando el

	var	rel.inf
HP_avg	HP_avg	49.777663890
HST_avg	HST_avg	28.747074739
HomeTeam	HomeTeam	13.827177376
AwayTeam	AwayTeam	3.708578413
B365H	B365H	1.536926242
B365A	B365A	0.833072161
HC_avg	HC_avg	0.560249534
B365D	B365D	0.404362654
HS_avg	HS_avg	0.219384279
HF_avg	HF_avg	0.203510653
HY_avg	HY_avg	0.132233734
AC_avg	AC_avg	0.019818339
AS_avg	AS_avg	0.014913239
HR_avg	HR_avg	0.008316227
AST_avg	AST_avg	0.006718519
AF_avg	AF_avg	0.000000000
AY_avg	AY_avg	0.000000000
AR_avg	AR_avg	0.000000000
AP_avg	AP_avg	0.000000000



principio de analogía. Se obtienen en el mismo orden exactamente las mismas variables que el caso anterior estudiado, por tanto, las conclusiones que se extraen son las mismas.

A la hora de medir la capacidad de predicción del conjunto de entrenamiento y de prueba, se han obtenido los siguientes resultados:

MSE.home.test	0.192019110853039
MSE.home.train	0.190427513881156

Continuamos sin tener sobreajuste, y además se ha conseguido obtener mediante un método de mejora del rendimiento el mejor modelo obtenido dentro de los desarrollados. Justificando todos los hechos de razonamiento seguidos también para esta variable a predecir.

Conclusiones del análisis de regresión

A continuación, se van a comparar los cinco modelos de regresión efectuados hasta la fecha para tratar de predecir la variable salario del conjunto de datos. La elección se va a realizar fijándonos en

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

la capacidad de predicción de los modelos tanto para los conjuntos de entrenamiento y de test, fijándome en el error cuadrático medio cometido con cada conjunto de datos.

Variable	Modelo	MSE (train)	MSE (test)
Local	Regresión lineal multivariable	0.313	0.305
Visitante	Regresión lineal multivariable	0.209	0.206
Local	Lasso-Ridge	0.262	0.258
Local	Red neuronal	0.250	0.298
Visitante	Red neuronal	0.285	0.315
Local	Máquina de soporte virtual	0.264	0.279
Visitante	Máquina de soporte virtual	0.428	0.439
Local	Boositing	0.192	0.190
Visitante	Boosting	0.192	0.191

En la tabla podemos apreciar la ausencia de sobreajuste en todos los modelos desarrollados. Pese a que la complejidad de los modelos ha variado en los modelos desarrollados, este hecho se pude plasmar en las diferencias en el MSE, el hecho de haber trabajo con un conjunto de datos tan equilibrio nos ha proporcionado suficiente estabilidad para despreocuparnos del sobreajuste. Este hecho seguramente se deba a la acción explicada en la introducción del trabajo, donde para determinar las variables del conjunto de prueba se emplearon datos anteriores, aportando robustez al conjunto de prueba.

A la hora de elegir entre un modelo u otro, me voy a fijar en su rendimiento a la hora de predecir los datos de prueba. Si nos fijamos en el modelo el menor MSE identificamos a Boosting como el mejor modelo desarrollado.

Considero que ha sido adecuado el procedimiento a la hora de realizar los modelos explicados. Por último, cabe mencionar aquellos hechos relacionados con las variables explicativas, donde prevalecen las variables que están directamente relacionadas con la salida tanto porque son del mismo equipo (demostrado en los modelos de Regresión lineal y boosting), o porque hay una relación directa con marcar un gol (demostrado en boosting).

Comprobación con casos específicos

A la hora de analizar los casos específicos se empleó el mismo procedimiento que para la clasificación. En primer lugar, se seleccionaron cuatro encuentros de manera aleatoria donde únicamente determinamos el equipo local y el equipo visitante. Para determinar los atributos de dichos encuentros se empleó el mismo procedimiento empleado en el conjunto de entrenamiento (se realizó un promedio de los atributos explicativos de cada variable).

A continuación, se introdujeron dichos casos en el mejor modelo de regresión desarrollado, obteniéndose los resultados que se muestran a continuación:

Casos reales:

```
> round(test$FTHG_avg)
[1] 1 3 1 4
> round(test$FTAG_avg)
[1] 0 1 1 2
```

Predicción del número de goles del equipo local:

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

```
> salida_home
[1] 2 3 1 4
> caso_home
[1] 1.8188059 3.1360811 0.5910203 3.7717062
```

Predicción del número de goles del equipo visitante.

```
> salida_away
[1] 1 1 1 1
> caso_away
[1] 0.6346161 1.4586250 1.3148260 1.0660053
```

Como se puede comprobar, en el caso de los goles marcados por el equipo local se acertaron 4 resultados. Y en el caso del equipo visitante dos resultados. Esta comprobación no es eficaz para determinar que un modelo sea mejor que otro, ya que solo se han puesto 4 casos específicos. Aun así, sirve bastante para ver la utilidad del trabajo desarrollado.

Clasificación vs Regresión

Con el fin de poner en común los conocimientos extraídos tanto por María como por Ignacio, se extrapoló el conocimiento extraído mediante el mejor modelo de regresión para compararlo con el mejor modelo de clasificación.

Este hecho se realizó porque tu al predecir el número de goles que marca el equipo local y el equipo visitante puedes determinar qué equipo ganó, perdió o empato. Por tanto, tras redondear la salida del modelo de regresión a valores discretos, acción que no se había hecho para medir en mejor medida la eficacia del error cometido, se puede obtener una salida propia de un modelo de clasificación.

La comparación de ambos modelos se realizó en el conjunto de datos de prueba. Se escogieron los mejores algoritmos dentro de cada uno de los análisis realizados.

Recordamos que el mejor modelo de clasificación es el siguiente:

Un algoritmo KMeans con 3 clústers y una precisión de 0,492.

```
> table_pred<-table(test.labels, prediction)
> table_pred
           prediction
test.labels   1   2   3
          A  63   8  31
          D  81  15  14
          H 107  49  12
```

Al extrapolar el conocimiento del análisis de regresión se obtuvo la siguiente matriz de confusión.

```
   hola
     A  D  H
  A 18 52 32
  D 12 58 40
  H 13 71 84
> table <- table(liga_test$FTR, hola)
> accuracy <- round((sum(diag(table))/sum(table))*100,2)
> accuracy
[1] 42.11
```

Como se puede apreciar, el intento ha sido en vano, la precisión obtenida es considerablemente inferior al 0'5, el cual es el mejor valor obtenido para el mejor estudio de clasificación. Por tanto, una de las conclusiones que podemos extraer de esta puesta en común si perseguimos un fin específico, debemos de utilizar los modelos que procedan.

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Otra conclusión adicional es que un mismo problema se puede tratar de resolver desde diferentes perspectivas, siempre y cuando haya un punto en común coherente que nos permita dicha comparación.

Conclusiones

Objetivos cubiertos

En general, podemos decir que estamos satisfechos con el cumplimento de los objetivos marcados. Hemos utilizado tanto técnicas de regresión (Ignacio) y de clasificación (María) para plantear el análisis de los resultados de partidos. Hemos aprendido que variables son más significativas a la hora de predecir tanto los resultados como los goles de cada encuentro, y por lo tanto cocemos que características del juego son más determinantes, uno de nuestros objetivos marcados.

Reflexiones

Este trabajo demuestra que predecir el fútbol sigue siendo una tarea difícil. En el deporte intervienen multitud de aspectos, muchos de ellos no medibles ni cuantificables, siendo precisamente esto la esencia no sólo del fútbol. El fútbol es un deporte compuesto por personas y, por ello, se encuentra subordinado a las limitaciones y errores de los seres humanos: estados de ánimo, estados físicos, aspectos extradeportivos ....

No obstante, podemos ver, a raíz de este trabajo, que los algoritmos de machine learning son capaces de "pensar" de manera más precisa que una persona que no tenga idea de fútbol. La ventaja en la predicción de estos algoritmos frente a una predicción aleatoria supone un 17% de mejora.

Una de las cosas más importantes de la que nos hemos dado cuenta es la importancia de tener los datos correctos. Antes de emplear la media de los tres últimos partidos como variables predictoras, empleamos directamente los datos de cada partido recogidos a posteriori. Con estos últimos obteníamos modelos más precisos, pero ahora sabemos que era una falsa ilusión, pues solo nos permitía predecir partidos pasados. Con los datos correctos, podemos predecir partidos futuros, y aunque la precisión sea menor, pensamos que saber lo que va a ocurrir en un partido futuro es de mucho más valor y tiene muchas más aplicaciones prácticas que fijarse solo en el pasado.

Mejoras

Durante la realización del trabajo hemos pensado en algunas mejoras que podrían hacerse. Para una mayor precisión creemos que sería interesante investigar las siguientes variables: los jugadores de cada equipo que disputan el partido, la hinchada que acude al encuentro o las condiciones climáticas.

Además, en lugar de utilizar la media de los datos de tres últimos partidos para todas las variables, podíamos haber realizado un estudio diferente para cada variable, por ejemplo, utilizando una media ponderada para los goles, dando más importancia al partido más cercano en el tiempo al nuestro, o utilizando solo el número de tarjetas del último partido en lugar de los tres anteriores.

Aplicaciones prácticas

María Oliva Calero
Ignacio García Santamaría
3ºB GITT

Hemos reflexionado sobre las posibles aplicaciones prácticas que podría tener este trabajo. Aunque nuestro trabajo nace por simple curiosidad sobre como lo estudiado este cuatrimestre podría a aplicarse a un campo que nos atrae, como es el fútbol y en concreto, La Liga, hemos llegado a la conclusión de que podría tener varias aplicaciones prácticas.

Probablemente no en un futuro muy próximo, pero creemos que la investigación de datos en el deporte, y en concreto en el fútbol, es un campo con muchísimo potencial aún por descubrir. Éste trabajo solo refuerza esta opinión: machine learning en el deporte es un campo en desarrollo. Si modelos relativamente sencillos como los nuestros consiguen estos resultados, esperamos ver en el futuro modelos mejores y más complejos.

En concreto nuestro trabajo se podría utilizar en el campo de las apuestas deportivas, para obtener beneficios apostando a los resultados de los partidos. En un futuro, podríamos incluso llegar a predecir el número de tarjetas por partido o el número de córneres, aspectos que pueden aumentar considerablemente la cuota de una apuesta deportiva, y por tanto, reportar más beneficio.