Redes neuronales

Efrén Jiménez

1 de setiembre de 2016

Análisis del Problema

El deporte, como inustria, mueve millones de dólares al año. Con respecto a las contrataciones, diferentes tienen diferentes formas de "agrupar" o "catalogar" a los jugadores, pero en general hay una categoría de jugadores sumamente cotizados que son considerados "súper estrellas". Las implicaciones de estos jugadores son muchas, pues no solo representan una gran inversión para el equipo sino que también se convierten en una de las razones principales por las cuales el público va al estadio. A nivel de mercadotécnia, también es una decsisión súmamente importante pues las súper estrellas suelen representar una de las principales cartas comerciales.

Por estas razones, la decisión de si un jugador puede ser o no una súper estrella es de suma importancia, y tener un modelo que pueda predecir potenciales súper estrellas para asisiter en la toma de estas decisiones puede ser de suma importancia para diferentes equipos.

Entendimiento de los Datos

Para analizar este problema, se tiene un conjunto de datos con 263 observaciones y 19 variables:

- Nombre del jugador: cualitativa; con 263 valores diferentes.
- ID de la Posición: cualitativa; con 10 valores diferentes entre 0 y 10.
- Tiros: cuantitativa; cantidad de tiros en la temporada anterior, con un rango entre 19 y 687.
- Tiros Hechos: cuantitativa; cantidad de tiros anotados en la temporada anterior, con un rango entre 1 y 238.
- Puntos Personales: cuantitativa; puntos anotados personalmente por el atleta la temporada anterior, con un rango entre 0 y 40.
- Puntos Totales: cuantitativa; cantidad de puntos que el atleta contribuyó a anotar, con un rango entre 0 y 130.
- Asistencias: cuantitativa; cantidad de veces que el atleta contribuyó a quitarle la posesión del balón al rival, con un rango entre 0 y 121.
- Concesiones: cuantitativa; cantidad de veces que una jugada del atleta causó que el rival cediera una ventaja a la ofensiva, con un valor entre 0 y 105.
- Bloqueos: cuantitativa; cantidad de veces que el atleta bloqueó a un rival la temporada anterior, con un valor entre 0 y 1377.
- Asistencias a Bloqueos: cuantitativa; cantidad de veces que el atleta asistió a un compañero a bloquear a un rival, con un rango entre 0 y 492.
- Faltas: cuantitativa; cantidad de veces que el atleta cometió una falta, con un rival entre 0 y 32.
- Años de profesional: cuantitativa; cantidad de año que el atleta ha jugado a nivel profesional, con un rango entre 1 y 24.
- Tiros en la carrera: cuantitativa; cantidad de tiros en toda la carrera del atleta, con un rango de 19 a 14053.
- Tiros hechos en la carrera: cuantitativa; cantidad de tiros hechos en la carrera del atleta, con un rango de 4 a 4256.
- Puntos Personales en la carrera: cuantitativa; cantidad de puntos personales en la carrera del atleta, con un rango de 0 a 548.
- Puntos Totales en la carrera: cuantitativa; puntos totales del atleta en toda su carrera, con un rango de 2 a 2165.

- Asistencias en la carrera: cuantitativa; cantidad de asistencias en toda la carrera, con un rango de 3 a 1659.
- Concesiones en la carrera: cuantitativa; cantidad de concesiones en toda la carrera, con un rango de 1 a 1566.
- Valor para el equipo: cualitativa; valor que puede tener el jugador para el equipo, con cuatro valores posibles: Jugador de Rol, Contribuidor, Jugador de Franquicia y Súper Estrella. Esta variable se manipuló durante el análisis para convertirla en binaria, respondiendo a la pregunta: ¿Es el jugador una súper estrella? donde 0 es 'No' y 1 es 'Sí'.

Exploración de los Datos

Antes de explorar los datos, es necesario cargarlos y aplicar la transformación respectiva a la variable "valor para el equipo", pues el problema consiste en determinar cuáles jugadores son potencialmente súper estrellas.

Adicionalmente, se procede a dividir el conjunto de datos en uno de entrenamiento y otro de prueba.

```
library(caTools)
library(neuralnet)
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.
```

```
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

##
## Attaching package: 'ROCR'

## The following object is masked from 'package:neuralnet':
##
##     prediction
```

```
library(pROC)
```

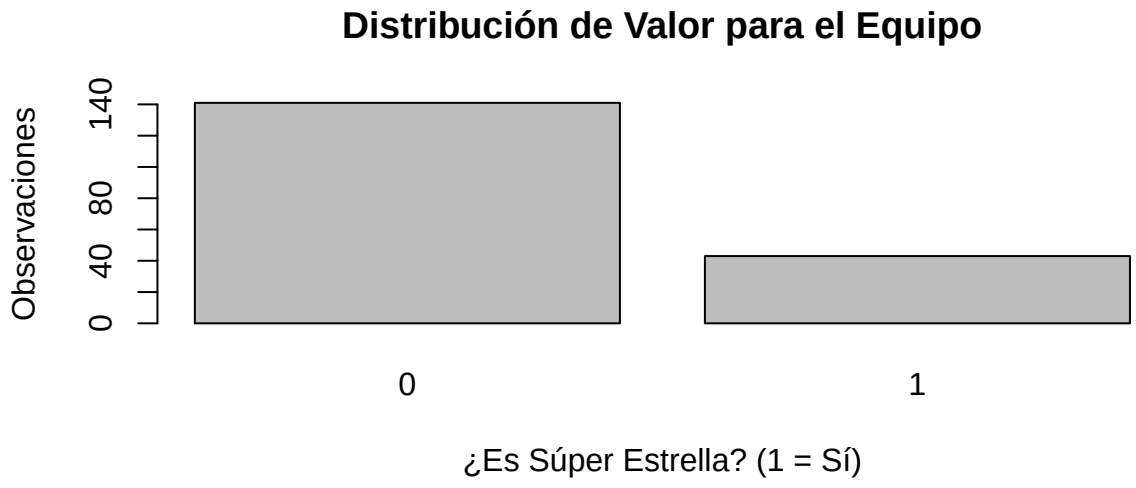
```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
setwd('D:\\Drive\\Universidad\\UTN\\2016\\III Cuatrimestre\\mineria_2016_III_cuatri\\Clase 12\\Redes Ne

datos <- read.csv('datos.csv')

datos$Team_Value <- as.character(datos$Team_Value)

datos[datos$Team_Value != 'Superstar',]$Team_Value <- 0
datos[datos$Team_Value == 'Superstar',]$Team_Value <- 1

datos$Team_Value <- factor(datos$Team_Value, levels = c('0','1'))

splt <- sample.split(datos$Team_Value, SplitRatio = 0.7)

entrenamiento <- datos[splt, ]
prueba <- datos[!splt, ]
```

Como podemos observar, la mayoría de las observaciones no son súper estrellas. Esto es esperado, pues la esta categoría de jugadores generalmente son los menos.

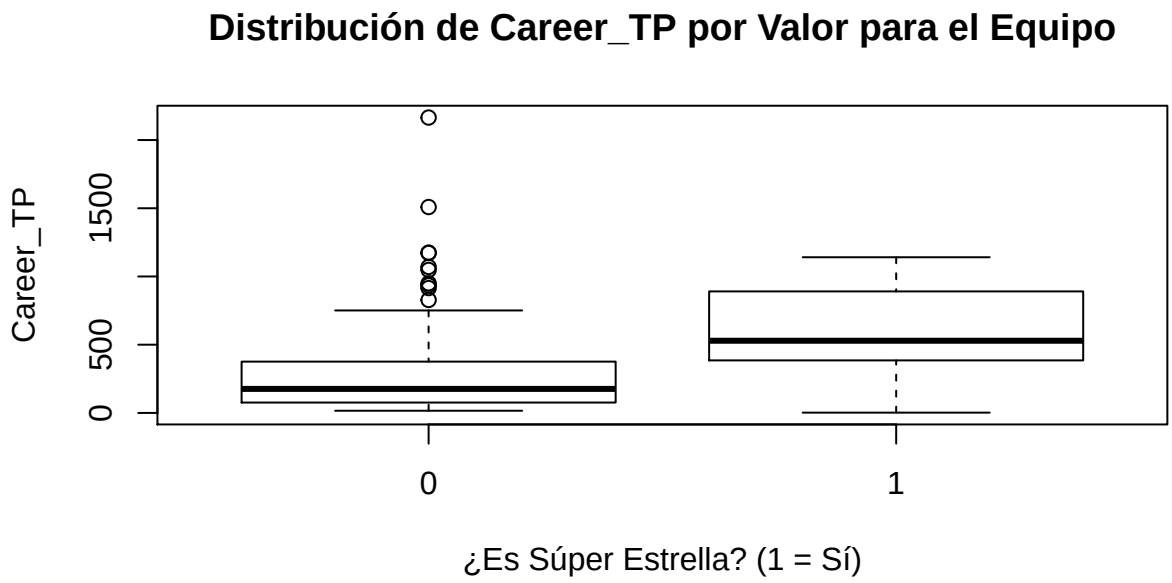
```
barplot(table(entrenamiento$Team_Value),
        main = 'Distribución de Valor para el Equipo',
        ylab = 'Observaciones',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



Para efectos de determinar qué variables influencian la determinación de si es una súper estrella o no, podemos analizar algunas de las variales que acumulan el desempeño de la carrera.

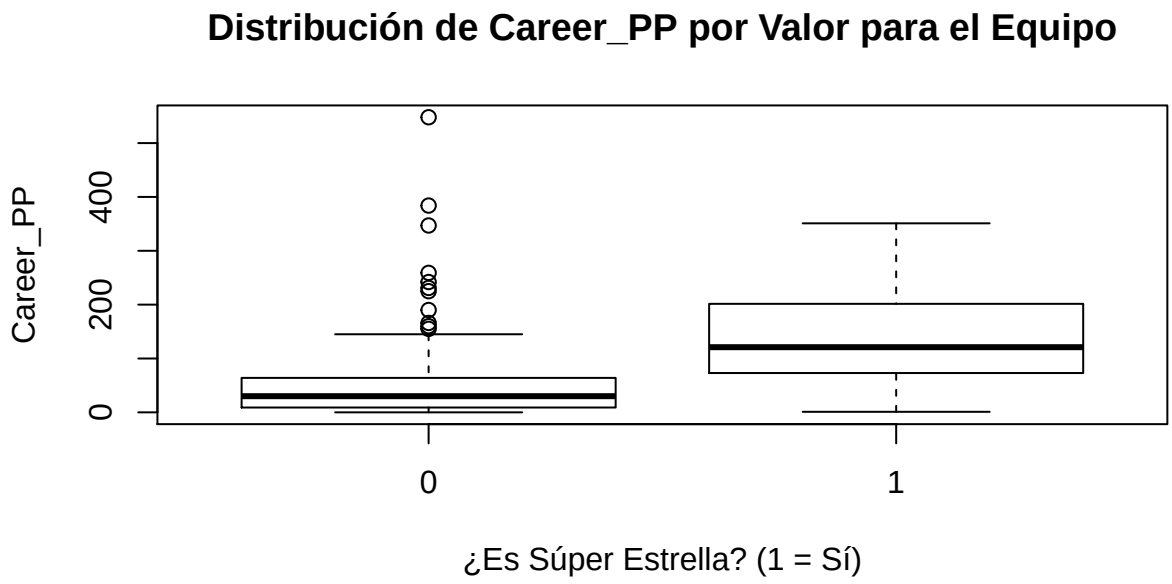
En el caso de la variable Career_TP, se puede apreciar una diferencia considerable entre las dos categorías. Sin embargo, los valores extremos de la categoría 'No' o 0 son mayores. Estos puntos anómalos en esa categoría pueden ser los jugadores de franquicia.

```r
boxplot(Career_TP ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Career_TP por Valor para el Equipo',
        ylab = 'Career_TP',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



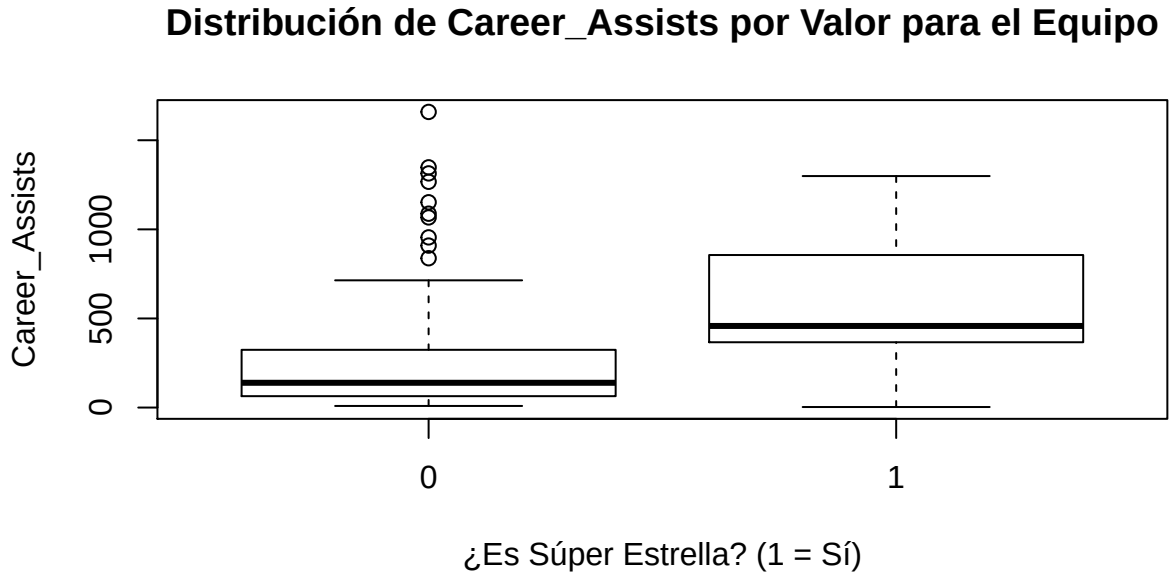
La variable Career_PP se comporta de manera similar:

```r
boxplot(Career_PP ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Career_PP por Valor para el Equipo',
        ylab = 'Career_PP',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



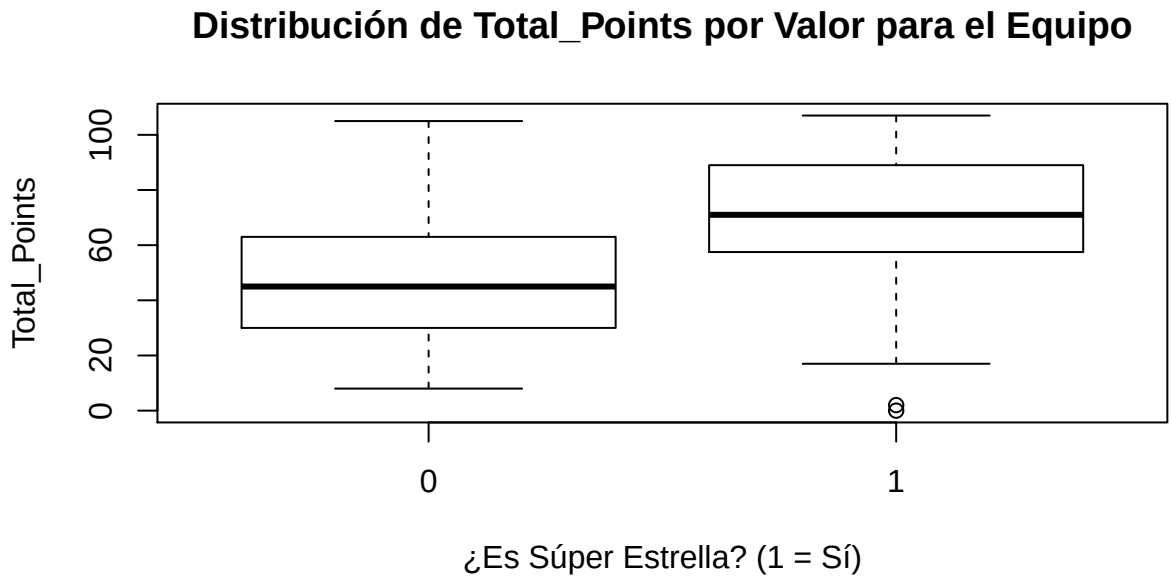
Y también la variable Career_Assists:

```
boxplot(Career_Assists ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Career_Assists por Valor para el Equipo',
        ylab = 'Career_Assists',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



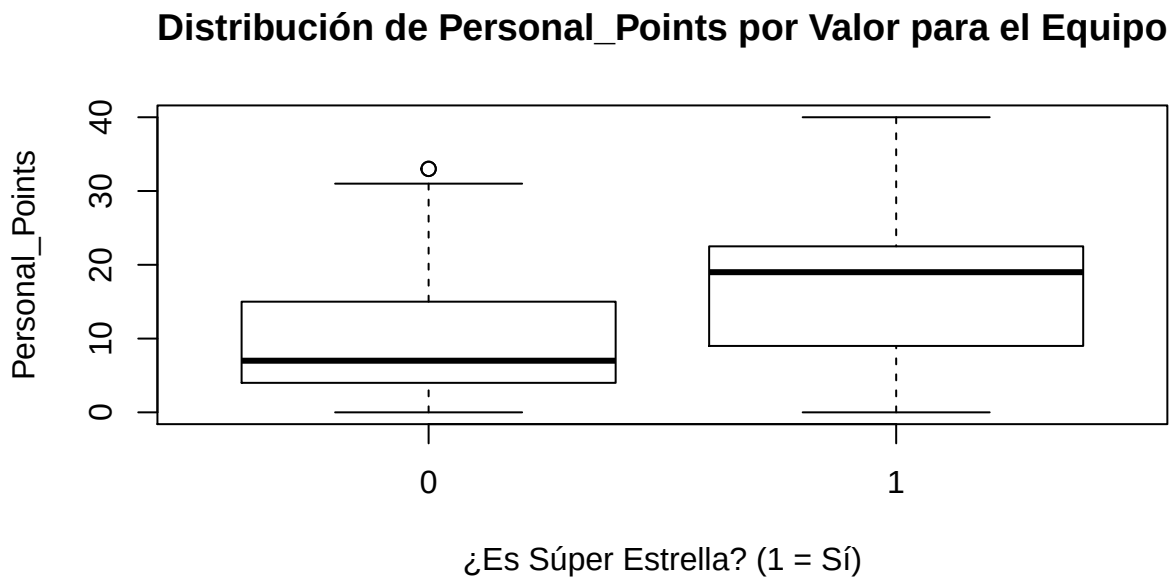
Cuando se analizan las variables a nivel de la temporada anterior, el patrón se mantiene, aunque las diferencias son menores:

```r
boxplot(Total_Points ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Total_Points por Valor para el Equipo',
        ylab = 'Total_Points',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



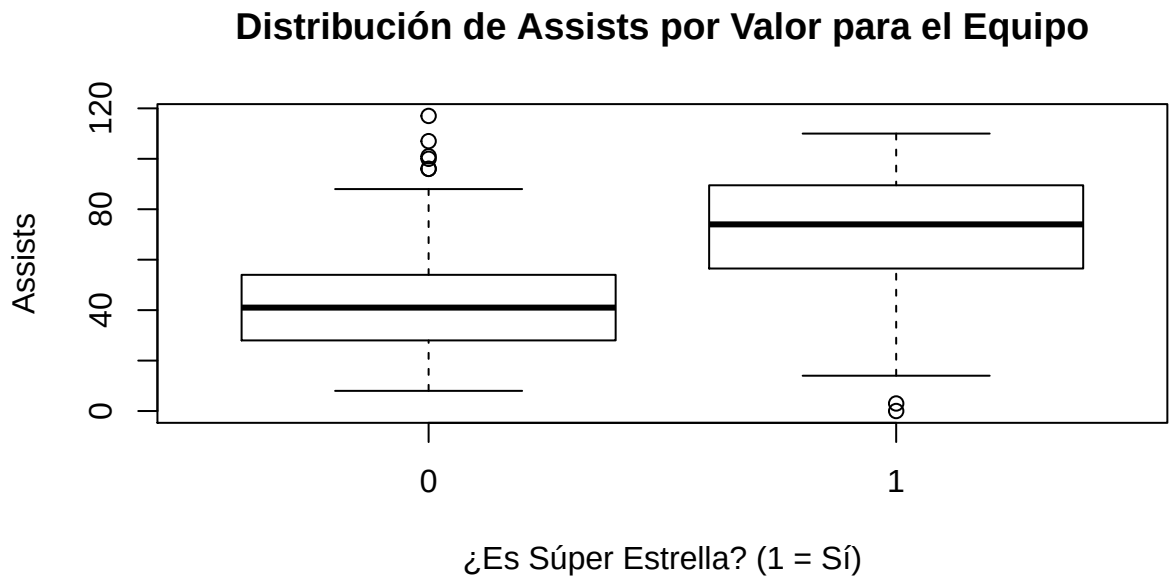
La variable Career_PP se comporta de manera similar:

```r
boxplot(Personal_Points ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Personal_Points por Valor para el Equipo',
        ylab = 'Personal_Points',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



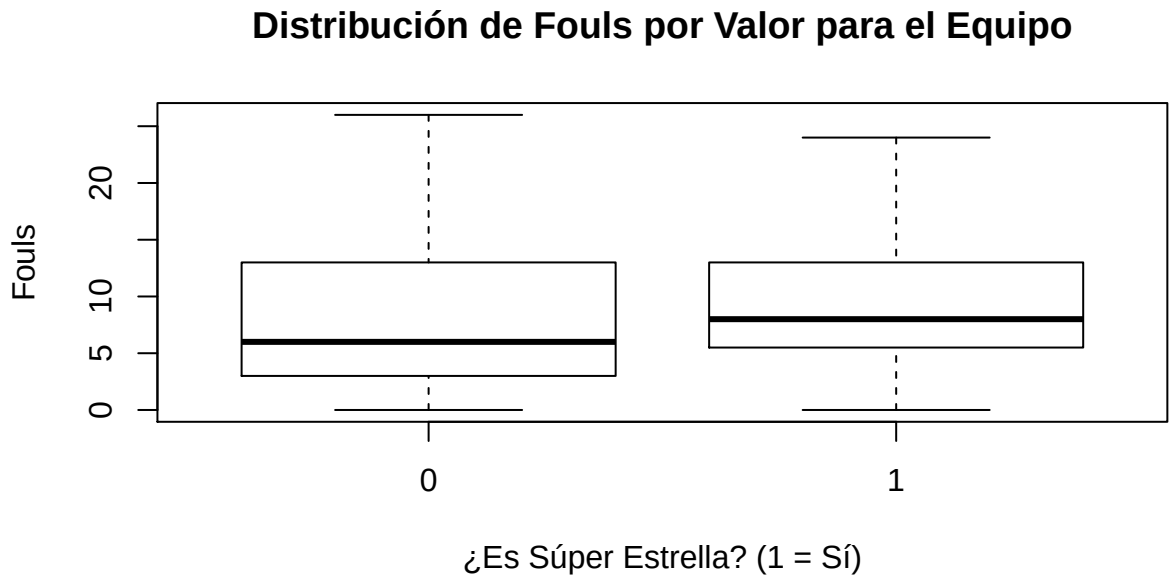
Y también la variable Career_Assists:

```
boxplot(Assists ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Assists por Valor para el Equipo',
        ylab = 'Assists',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



Hay algunas variables que tienden a favorecer el no, como por ejemplo la cantidad de faltas la temporada anterior:

```r
boxplot(Fouls ~ Team_Value,
        data = entrenamiento,
        main = 'Distribución de Fouls por Valor para el Equipo',
        ylab = 'Fouls',
        xlab = '¿Es Súper Estrella? (1 = Sí)')
```



A pesar de que el análisis exploratorio nos indica que podríamos guiarnos nada más por las estadísticas a nivel de la carrera, si hacemos esto, corremos el riesgo de no tomar encuenta súper estrellas potenciales que están en etapas tempranas en su carrera. Para evitar esto, y para aprovechar también la capacidad de las redes neuronales, vamos a tomar en cuenta todas las variables cuantitativas para intentar responder la pregunta.

Creación de los Modelos

En este caso, se van a utilizar una red neuronal, un árbol de decisión y un bosque aleatorio. Se comienza por crear la red neuronal y por aplicarla a los datos de prueba:

```r
#crear matrices numéricas para ser consumidas por las redes neuronales.
entrenamiento.red <-  model.matrix(~ Shots + Makes + Personal_Points + Total_Points + Assists + Concessi
                                   data = entrenamiento)

prueba.red <-  model.matrix(~ Shots + Makes + Personal_Points + Total_Points + Assists + Concessions + I
                                   data = prueba)

#ajustar los nombres de las columnas en las matrices
colnames(entrenamiento.red) <- make.names(colnames(entrenamiento.red))
colnames(prueba.red) <- make.names(colnames(prueba.red))

set.seed(14234)
#crear red neuronal con 7 unidades en la capa oculta
modelo.red <- neuralnet(Team_Value1 ~ Shots + Makes + Personal_Points + Total_Points + Assists + Concess
                        data = entrenamiento.red,
```

```
                        hidden = 7)

#realizar predicciones
predicciones.red <- compute(modelo.red, prueba.red[ , c(3:ncol(prueba.red) - 1)])
```

Posteriormente, se crea el árbol de decisión:

```
#crear modelo
set.seed(14234)
modelo.arbol <- rpart(Team_Value ~ Shots + Makes + Personal_Points + Total_Points + Assists + Concessio
                        data = entrenamiento)

#realizar predicciones
predicciones.arbol <- predict(modelo.arbol, newdata = prueba, type = 'prob')
```

Finalmente, el bosque aleatorio:

```
#crear modelo
set.seed(14234)
modelo.bosque <- randomForest(Team_Value ~ Shots + Makes + Personal_Points + Total_Points + Assists + C
                        data = entrenamiento)

#realizar predicciones
predicciones.bosque <- predict(modelo.bosque, newdata = prueba, type = 'prob')
```

Evaluación

El primer punto de comparación es contra un modelo ingenuo: (pronostica siempre 'no')

```
modelo.ingenuo <- rep(0, nrow(prueba))

table(prueba$Team_Value, modelo.ingenuo)
```

```
##    modelo.ingenuo
##       0
##   0 60
##   1 19
```

```
detach("package:neuralnet", unload=TRUE) #descargar la librería neural net para poder usar la función p

prediccionROC.ingenuo <- prediction(modelo.ingenuo, prueba$Team_Value)
as.numeric(performance(prediccionROC.ingenuo, "auc")@y.values)
```

```
## [1] 0.5
```

Métricas del Modelo Ingenuo:

- Exactitud: 75.95%
- Sensibilidad: 0%
- Especificidad: 100%

- Área bajo la curva: 50%

Posteriormente, averiguamos el desempeño de la red neuronal:

```
table(prueba$Team_Value, predicciones.red$net.result >= 0.5)
```

```
##
##     FALSE TRUE
##   0    59    1
##   1    19    0
```

```
prediccionROC.red <- prediction(predicciones.red$net.result, prueba$Team_Value)
as.numeric(performance(prediccionROC.red, "auc")@y.values)
```

```
## [1] 0.7219298246
```

Métricas de la red neuronal:

- Exactitud: 73.42%
- Sensibilidad: 0%
- Especificidad: 96.67%
- Área bajo la curva: 63.64%

Verificamos también el desempeño del árbol de decisión:

```
table(prueba$Team_Value, predicciones.arbol[,2] >= 0.5)
```

```
##
##     FALSE TRUE
##   0    51    9
##   1     4   15
```

```
prediccionROC.arbol <- prediction(predicciones.arbol[,2], prueba$Team_Value)
as.numeric(performance(prediccionROC.arbol, "auc")@y.values)
```

```
## [1] 0.8048245614
```

Métricas del árbol de decisión:

- Exactitud: 88.61%
- Sensibilidad: 57.89%
- Especificidad: 98.33%
- Área bajo la curva: 78.29%

Finalmente, vemos el desempeño del bosque aleatorio:

```
table(prueba$Team_Value, predicciones.bosque[,2] >= 0.5)
```

```
##
##     FALSE TRUE
##   0    55    5
##   1     8   11
```

```r
prediccionROC.bosque <- prediction(predicciones.bosque[,2], prueba$Team_Value)
as.numeric(performance(prediccionROC.bosque, "auc")@y.values)
```

```
## [1] 0.9114035088
```

Métricas del bosque aleatorio:

- Exactitud: 79.7589.87%
- Sensibilidad: 63.16%
- Especificidad: 98.33%
- Área bajo la curva: 93.60%

Nota: a pesar de haber incluido la instrucción set.seed, al generar el documento pdf los resultados varían. Por este motivo, es posible que las tablas resumenes y el área bajo la curva mencionados anteriormente se vean diferentes.

Resultados

Luego de crear y analizar diferentes modelos, podemos sacar las siguientes conclusiones:

- La red neuronal creada en este caso en general tiene resultados muy similares (a veces peores) que el modelo ingenuo.
- El algoritmo con el mejor desempeño generalmente es el de bosques aleatorios.
- En algunas iteraciones, la exactitud del bosque aleatorio era inferior a la del árbol de decisión, pero su sensibilidad se mantenía mejor. En dichos casos, dado el hecho de que para el negocio es más importante identificar apropiadamente a una super estrella versus la exactitud en aquellos que no lo son, la recomendación hubiera sido seguir al modelo con mayor sensibilidad en lugar del de mayor exactitud.