



Scouting futbol

Proyecto final Data Science - Coderhouse





Abstract

Uno de los problemas a que se enfrentan los equipos de fútbol es la elección de los refuerzos para determinadas posiciones que formarán parte de la plantilla. Esta elección se realiza tomando en cuenta dos aspectos: Costos de la contratación de un jugador y el potencial técnico que posee.

El siguiente proyecto intenta brindar una ayuda hacia los scouter de equipos de fútbol. A través del análisis de datos y herramientas de cientos de datos, se buscará aportar con nombres de jugadores como sugerencia de adquisiciones, en base a cierto presupuesto.





Preguntas

1

¿Cómo podemos categorizar a los jugadores para contratar aquellos que pueden brindar una mayor rentabilidad de la inversión, en base al mayor potencial técnico que representan?

2

Para determinado presupuesto y posición, ¿podremos tener una herramienta que nos proporcione un listado de jugadores candidatos a contratar?

3

¿Cuáles son las variables (condiciones técnicas) que explican de mejor manera el rendimiento y el potencial de un jugador?



Objetivo del proyecto

El siguiente proyecto intenta brindar una ayuda hacia los scouter de equipos de fútbol.

A través del análisis de datos y herramientas de ciencias de datos buscaré poder predecir el nivel del jugador en base a sus estadísticas. Se buscará aportar con nombres de jugadores como sugerencia de adquisiciones, en base a cierto presupuesto.



Contexto de negocio

Uno de los problemas a que se enfrentan los equipos de fútbol es la elección de los refuerzos para determinadas posiciones que formarán parte de la plantilla. Esta elección se realiza tomando en cuenta dos aspectos: Costos de la contratación de un jugador y el potencial técnico que posee.

Data acquisition

- La siguiente base de datos se obtuvo de una API de fútbol <https://apifootball.com/>.
- Corresponde a las estadísticas (goles, atajadas, pases, entre otros) de todos los jugadores que jugaron en el año 2021.
- Contiene 59713 entradas para 9467 jugadores distintos.



API Football

Es una API que provee de información de fútbol de todo el mundo. Brinda datos de partidos, resultados, estadísticas, equipos, torneos y de jugadores.

#	Column	Non-Null Count	Dtype
0	player.id	59713	non-null float64
1	player.marketValue	10549	non-null float64
2	player.team.name	11443	non-null object
3	player.team.primaryUniqueTournament.name	10418	non-null object
4	player.name	59713	non-null object
5	position	59713	non-null object
6	player.country.name	59713	non-null object
7	minutesPlayed	59713	non-null float64
8	accurateCross	59713	non-null float64
9	accurateLongBalls	59713	non-null float64
10	accuratePass	59713	non-null float64
11	aerialLost	59713	non-null float64
12	aerialWon	59713	non-null float64
13	bigChanceCreated	59713	non-null float64
14	bigChanceMissed	59713	non-null float64
15	blockedScoringAttempt	59713	non-null float64
16	challengeLost	59713	non-null float64
17	dispossessed	59713	non-null float64
18	duelLost	59713	non-null float64
19	duelWon	59713	non-null float64
20	fouls	59713	non-null float64
21	goals	59713	non-null float64
22	interceptionWon	59713	non-null float64
23	keyPass	59713	non-null float64
24	onTargetScoringAttempt	59713	non-null float64
25	ownGoals	59713	non-null float64
26	outfielderBlock	59713	non-null float64
27	possessionLostCtrl	59713	non-null float64
28	punches	59713	non-null float64
29	rating	59713	non-null float64
30	savedShotsFromInsideTheBox	59713	non-null float64
31	saves	59713	non-null float64
32	shotOffTarget	59713	non-null float64
33	totalClearance	59713	non-null float64
34	totalContest	59713	non-null float64
35	totalCross	59713	non-null float64
36	totalLongBalls	59713	non-null float64
37	totalOffside	59713	non-null float64
38	totalPass	59713	non-null float64
39	totalTackle	59713	non-null float64
40	touches	59713	non-null float64
41	wasFouled	59713	non-null float64
42	wonContest	59713	non-null float64

EDA - Exploratory Data Analysis



*Minutos
jugados*



*Valor
mercado*



Posiciones

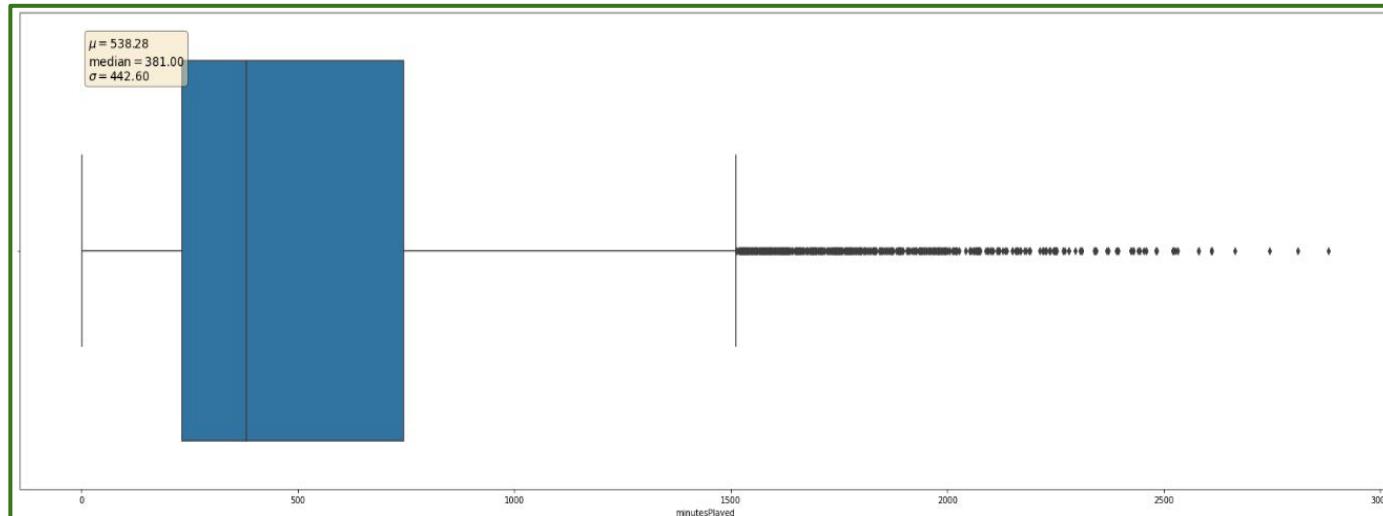


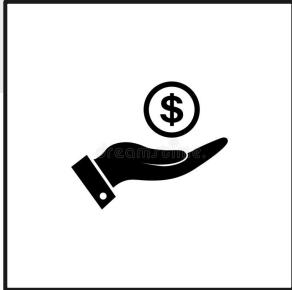
Correlaciones



1 data.describe().round(2)									bis
	player.id	player.marketValue	minutesPlayed	accurateCross	accurateLongBalls	accuratePass	aerialLost	aerialWon	bis
count	59713.00	1.054900e+04	59713.00	59713.00	59713.00	59713.00	59713.00	59713.00	59713.00
mean	734604.50	2.223908e+06	186.60	0.80	5.54	62.22	3.16	3.16	3.16
std	374700.91	7.141917e+06	365.42	3.43	15.87	144.00	9.14	9.87	9.87
min	80.00	0.000000e+00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	341091.00	1.400000e+05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50%	886429.00	3.150000e+05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
75%	1009147.00	1.100000e+06	264.00	0.00	2.00	54.00	0.00	0.00	0.00
max	1398910.00	1.700000e+08	2880.00	97.00	311.00	2463.00	231.00	281.00	281.00

Un cuarto de los jugadores tienen 0 minutos jugados en todo el año, por lo tanto todas sus estadísticas serán nulas. Estos jugadores no me sirven ya que no puedo hacer ningún análisis de alguien que nunca jugó.

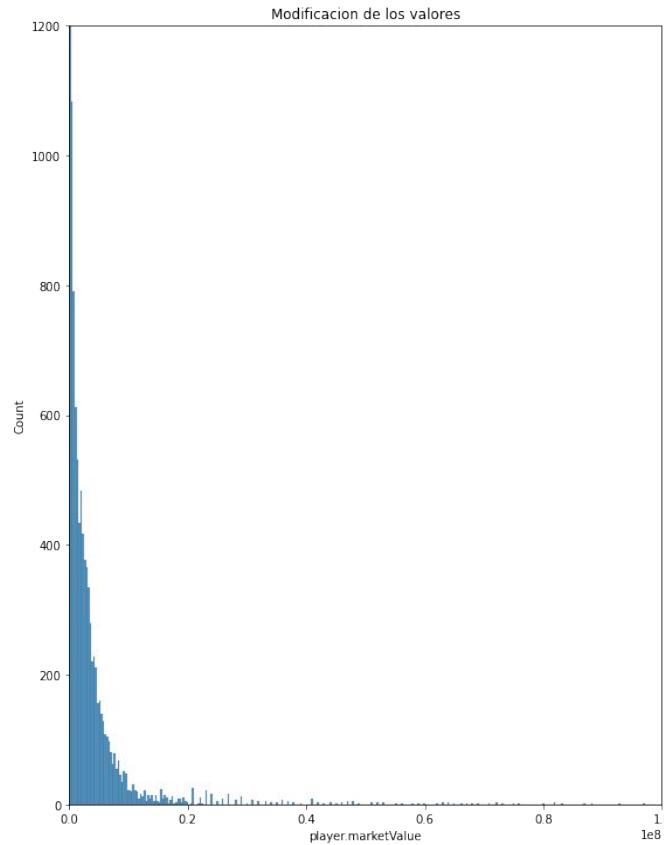
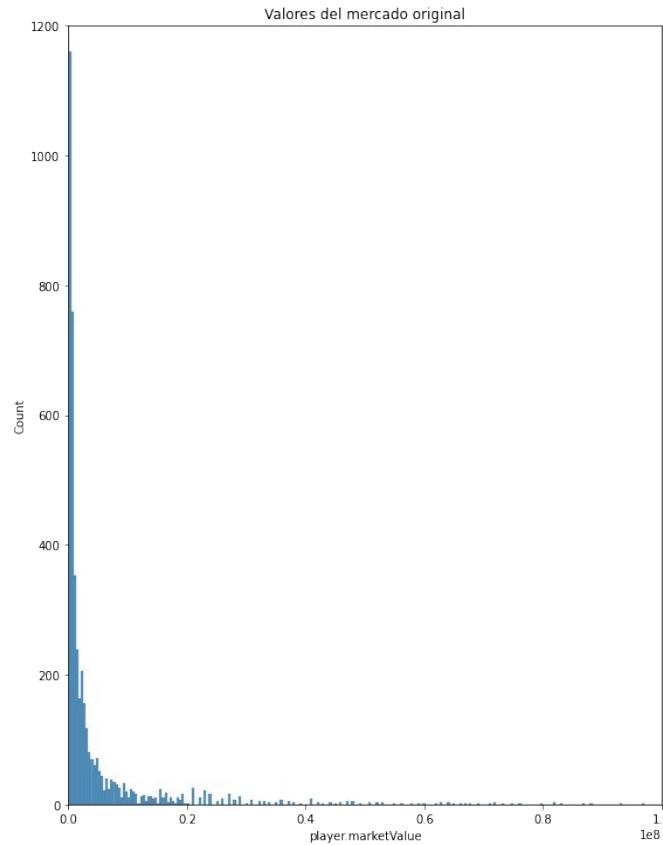




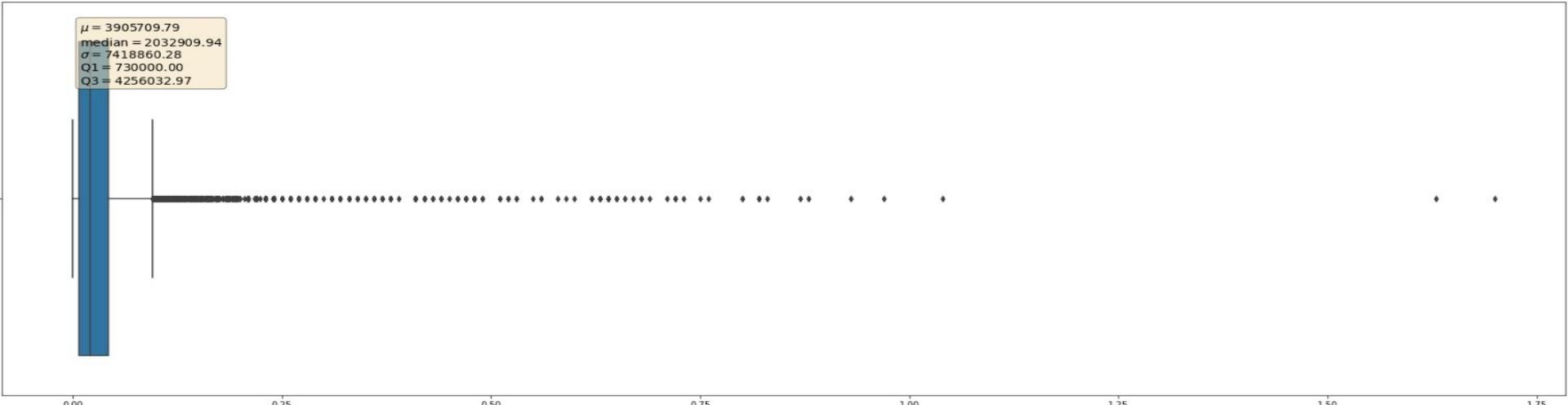
Observamos que hay 6029 jugadores que no tienen un valor en el mercado, que tienen valores nulos.

Implementamos imperative imputer, una herramienta predictiva para el valor del mercado del jugador en función de las variables.

Verificamos que no cambie la distribución de los datos.



$\mu = 3905709.79$
median = 2032909.94
 $\sigma = 7418860.28$
Q1 = 730000.00
Q3 = 4256032.97



50%

El 50% cobran entre €730.000 y
€4.256.033



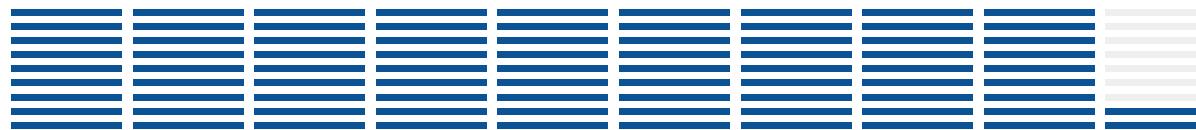
75%

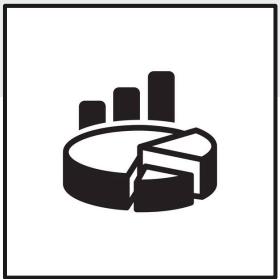
El 75% cobra menos de
€4.256.033



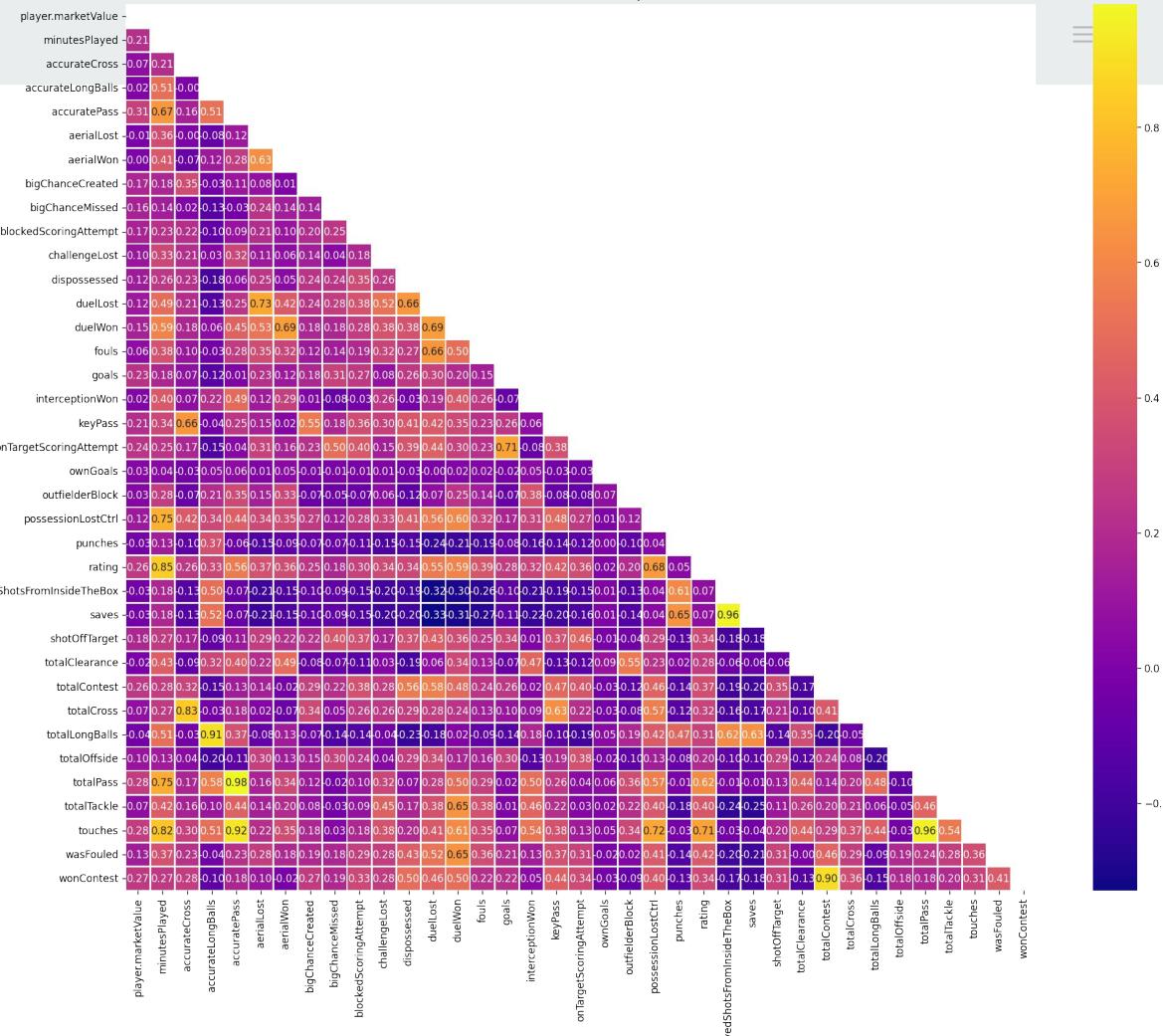
7%

Son jugadores con valores atípicos de
valor. Valiendo más que €9.545.082





Correlation Heatmap



- Minutos jugados y la cantidad de veces que la toca un jugador tiene correlaciones altas con varias variables ya que a mayor tiempo jugado más posibilidades de hacer más cosas como así mientras más la toca el jugador dará un mayor número de estadísticas.
 - La variable 'rating' que es el puntaje que se le puso al jugador por cada partido tiene altas relaciones con las que son 'total...', mientras más participe mejor puntaje tendrá.
 - Atajadas y atajadas dentro del área no tienen correlación con casi todas las demás variables ya que es una estadística sola de los arqueros que son un porcentaje muy bajo de los jugadores totales.
 - El valor del jugador no tiene correlación positiva alta con ninguna variable, no hay un factor clave a la hora de ponerle un precio al jugador.



A R Q U E R O S

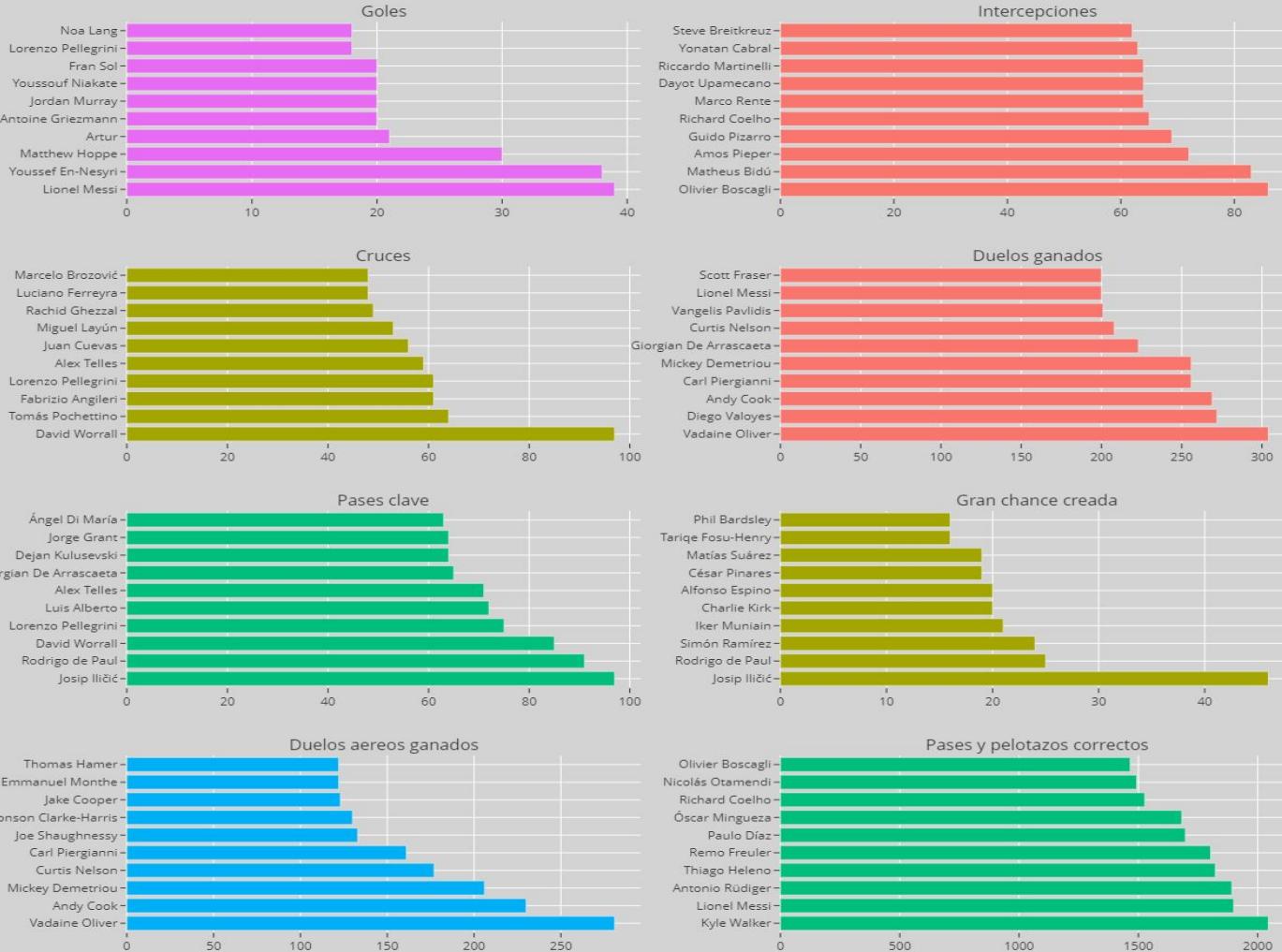
Top arqueros estadísticas

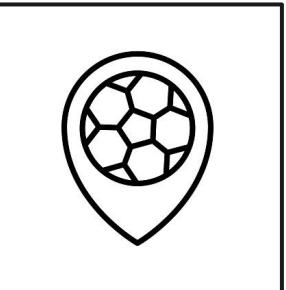


Top defensores estadísticas



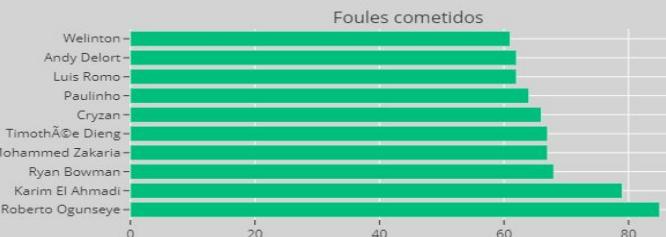
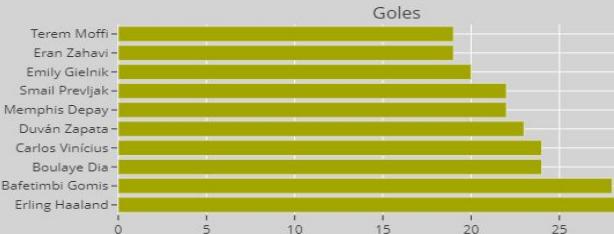
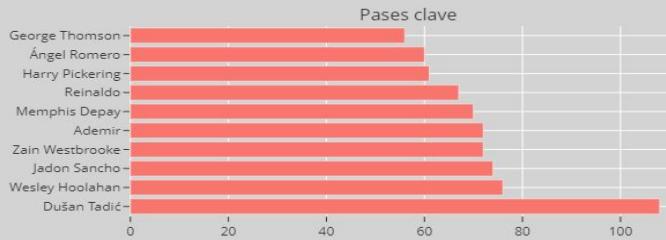
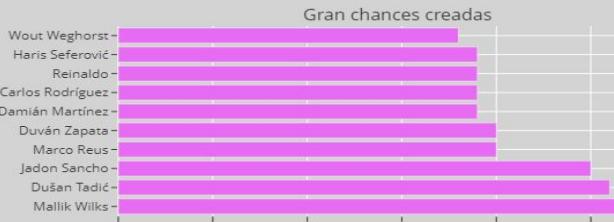
D
E
F
E
N
S
O
R
E
S

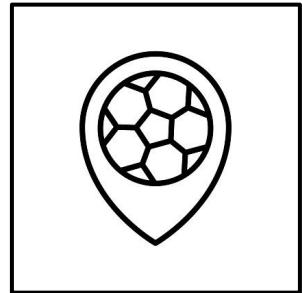




M E D I O C A M P I S T A S

Top mediocampistas estadísticas





DELANTEROS

Top delanteros estadísticas



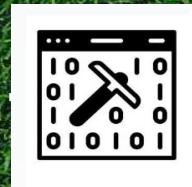
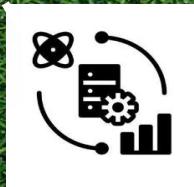
Algoritmos empleados

XGBoost



Ridge

Gradient Boosting



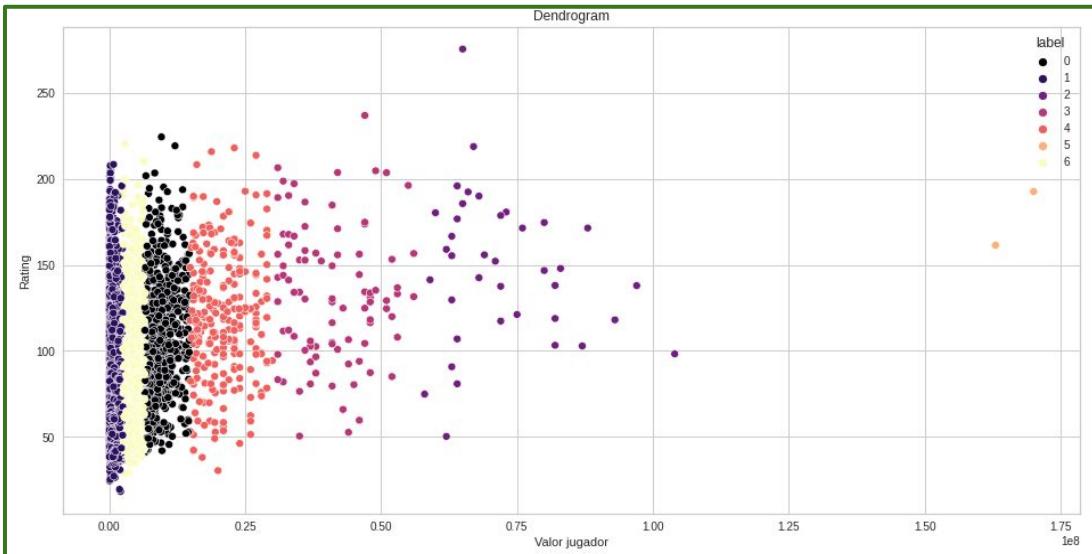
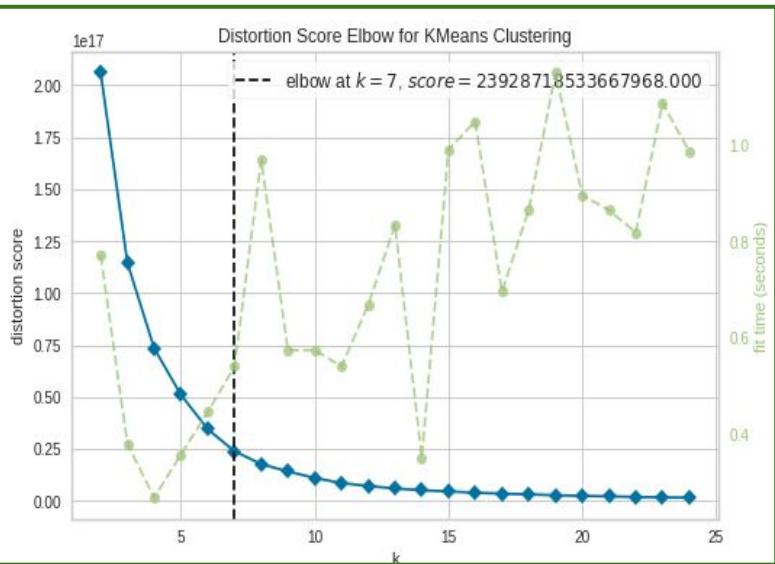
Iterative Imputer

K-Means



Clasificación con K-Means

Con K-means que es un algoritmo de clasificación no supervisada que agrupó jugadores en 7 grupos basándose entre su valor y su rating. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Cuando ya obtuvimos los 7 grupos los categorizamos con labels del 0 al 6. Al obtener la segunda imagen podemos determinar que vamos a trabajar con los jugadores de menores valores (grupos 0,1,4 y 6) ya que la idea es que el jugador sea de bajo valor.



Elección mejores variables

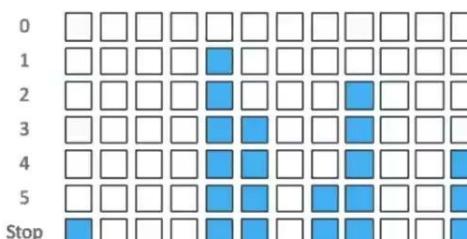
Con los métodos de feature selection seleccionaremos las variables que mejor definen el rating, valor que se le puso a cada jugador en cada partido. Para cada posición las variables serán distintas, ya que no le interesan puntuar a un arquero por la cantidad de goles que haga o a un delantero por cuantos quites hizo.

Los procesos que usaremos serán "Backward selection", "Forward selection" y "Stepwise". Estos se basan en un algoritmo de aprendizaje automático que lo encajaremos en nuestro data frame con las habilidades y con los objetivos. Siguen un enfoque de búsqueda codiciosa al evaluar todas las posibles combinaciones de habilidades contra el criterio de evaluación. Finalmente, selecciona la combinación de características que da el resultado óptimo para el algoritmo de aprendizaje automático especificado.

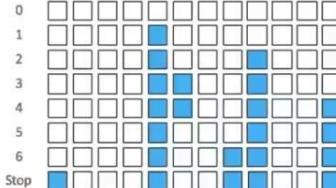
Backward Selection



Forward Selection



Mixed Selection (combination of forward and backward selection)



Elección de los algoritmos



Para elegir el mejor algoritmo de cada posición utilice el algoritmo llamado “Lazy predict”. Que hace esta herramienta? Ayuda a semi automatizar su tarea de aprendizaje automático, construye una gran cantidad de modelos básicos sin mucho código y ayuda a comprender qué modelos funcionan mejor sin ningún ajuste de parámetros.

Arqueros

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
HuberRegressor	0.97	0.97	5.79	0.02
SGDRegressor	0.97	0.97	5.79	0.07
Ridge	0.97	0.97	5.80	0.01
RidgeCV	0.97	0.97	5.80	0.03
BayesianRidge	0.97	0.97	5.80	0.01
Lars	0.97	0.97	5.80	0.01
LinearRegression	0.97	0.97	5.80	0.01
TransformedTargetRegressor	0.97	0.97	5.80	0.01
RANSACRegressor	0.97	0.97	5.80	0.03
LassoLarsIC	0.97	0.97	5.81	0.01
OrthogonalMatchingPursuitCV	0.97	0.97	5.83	0.04
ElasticNetCV	0.97	0.97	5.90	0.06
LassoCV	0.97	0.97	6.03	0.06
LinearSVR	0.97	0.97	6.04	0.01
LarsCV	0.97	0.97	6.04	0.02
LassoLarsCV	0.97	0.97	6.04	0.02
Lasso	0.97	0.97	6.12	0.01
XGBRegressor	0.96	0.96	6.35	0.17
GradientBoostingRegressor	0.96	0.96	6.48	0.06
HistGradientBoostingRegressor	0.96	0.96	6.52	0.26
ExtraTreesRegressor	0.96	0.96	6.64	0.14
PassiveAggressiveRegressor	0.96	0.96	6.68	0.02

Defensores

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ExtraTreesRegressor	0.84	0.85	12.39	0.98
RandomForestRegressor	0.84	0.84	12.42	1.77
HistGradientBoostingRegressor	0.84	0.84	12.50	6.51
XGBRegressor	0.84	0.84	12.50	0.26
LGBMRegressor	0.84	0.84	12.54	0.18
GradientBoostingRegressor	0.84	0.84	12.63	0.73
BaggingRegressor	0.83	0.83	13.02	0.34
RidgeCV	0.81	0.81	13.58	0.03
BayesianRidge	0.81	0.81	13.58	0.03
TransformedTargetRegressor	0.81	0.81	13.58	0.02
LinearRegression	0.81	0.81	13.58	0.04
Ridge	0.81	0.81	13.58	0.02
LassoCV	0.81	0.81	13.58	0.44
LarsCV	0.81	0.81	13.58	0.11
LassoLarsCV	0.81	0.81	13.58	0.13
LassoLarsIC	0.81	0.81	13.58	0.05
Lars	0.81	0.81	13.59	0.04

Mediocampistas

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
RandomForestRegressor	0.73	0.74	17.25	2.51
ExtraTreesRegressor	0.73	0.73	17.30	1.28
XGBRegressor	0.73	0.73	17.39	0.26
GradientBoostingRegressor	0.73	0.73	17.42	0.43
BaggingRegressor	0.72	0.73	17.45	0.70
LGBMRegressor	0.72	0.73	17.49	0.19
BaggingRegressor	0.72	0.72	17.59	0.26
LassoCV	0.72	0.72	17.68	0.14
LassoLarsCV	0.72	0.72	17.68	0.07
SGDRegressor	0.72	0.72	17.68	0.06
LassoLarsIC	0.72	0.72	17.68	0.05
RidgeCV	0.72	0.72	17.68	0.03
BayesianRidge	0.72	0.72	17.68	0.02
TransformedTargetRegressor	0.72	0.72	17.68	0.03
LinearRegression	0.72	0.72	17.68	0.03
Ridge	0.72	0.72	17.68	0.02
LassoCV	0.72	0.72	17.68	0.02
LassoLarsCV	0.72	0.72	17.68	0.02
LassoLarsIC	0.72	0.72	17.68	0.02
Lars	0.72	0.72	17.68	0.02
Lasso	0.72	0.72	17.72	0.18
ElasticNetCV	0.72	0.72	17.72	0.05
Lasso	0.72	0.72	17.73	0.03
OrthogonalMatchingPursuitCV	0.71	0.72	17.86	0.03
ExtraTreesRegressor	0.71	0.72	17.86	0.03
HuberRegressor	0.71	0.72	17.90	0.05
GradientBoostingRegressor	0.71	0.72	17.90	0.05
Lasso	0.71	0.72	17.90	0.05
PoissonRegressor	0.71	0.72	17.90	0.05

Delanteros

Model	Adjusted R-Squared	R-Squared	RMSE	Time Taken
ElasticNetCV	0.62	0.63	19.71	0.17
BayesianRidge	0.61	0.63	19.74	0.01
Ridge	0.61	0.63	19.75	0.02
RidgeCV	0.61	0.63	19.75	0.02
LassoLarsCV	0.61	0.63	19.75	0.10
LinearRegression	0.61	0.63	19.75	0.01
TransformedTargetRegressor	0.61	0.63	19.75	0.02
LassoLarsIC	0.61	0.63	19.76	0.03
LassoCV	0.61	0.63	19.76	0.17
SGDRegressor	0.61	0.62	19.82	0.04
RandomForestRegressor	0.61	0.62	19.86	1.17
XGBRegressor	0.61	0.62	19.95	0.19
OrthogonalMatchingPursuitCV	0.60	0.62	19.98	0.02
ExtraTreesRegressor	0.60	0.61	20.05	0.64
HuberRegressor	0.60	0.61	20.07	0.05
GradientBoostingRegressor	0.60	0.61	20.12	0.41
Lasso	0.60	0.61	20.13	0.02
PoissonRegressor	0.59	0.61	20.28	0.04
HistGradientBoostingRegressor	0.59	0.60	20.35	0.41
LarsCV	0.59	0.60	20.45	0.04



Resultados

Posición	Arquero	Defensor	Mediocampista	Delantero
Algoritmo	<u>Ridge</u>	<u>XGBoost</u>	<u>Gradient Boosting</u>	<u>Ridge</u>
<i>R2</i>	0.963	0.841	0.729	0.627
<i>MSE</i>	41.44	158	304.51	389.02
<i>RMSE</i>	6.44	12.57	17.45	19.73
<i>MAE</i>	4.83	8.35	13.33	15.47
<i>MAPE</i>	0.06	0.11	0.17	0.20
<i>MedAE</i>	3.92	5.63	10.93	13.30



Conclusiones

- Pudimos obtener unas listas con los mejores jugadores de las variables más importantes en cada posición.
- Las estadísticas nos sirven para ver realmente cómo está jugando el jugador y no por habilidades de un videojuego.
- La mejor predicción que podemos hacer es en los arqueros por las pocas cantidades de estadísticas que generan.
- Los delanteros y mediocampistas al ser tantas variables el modelo no es tan óptimo, se debería buscar algún método de reducción de dimensionalidad que pueda mejorar el algoritmo.
- La falta de datos del valor de mercado condiciona mucho a la hora de clasificar cuánto vale cada uno, por más que se hizo una imputación predictiva no es tan acertado.
- Con el modelo a medida que va generando estadísticas los jugadores podemos predecir con los minutos jugados si estará bien “puntuado”.
- La falta de datos como edad, equipo, liga, expected goals, entre otros ayudarían mucho más al modelo a la hora de hacer las predicciones.



Gracias!