Universidad de San Carlos de Guatemala
Facultad de ingeniería
Escuela de Estudios de Postgrado
Maestría en Ingeniería para la industria con Especialización en Ciencias de la Computación
Introducción al análisis de datos
Carlos Luis Pablo Hernández Gramajo.

Proyecto Documentación de procesos.

https://github.com/HernandezGramajo/Proyecto1_Aalisis_de_datos.git

Data Analisys

Se realizo el análisis de los datos para poder validar la información que se tiene se verificaron los 6 archivos palyer para ver si contenían la misma información.

Y posterior mente se realizo la carga de los archivos y se unificaron

```
Player_15 <- read.csv('../fifa2020/players_15.csv', sep=",")
Player_16 <- read.csv('../fifa2020/players_16.csv', sep=",")
Player_17 <- read.csv('../fifa2020/players_17.csv', sep=",")
Player_18 <- read.csv('../fifa2020/players_18.csv', sep=",")
Player_19 <- read.csv('../fifa2020/players_19.csv', sep=",")
Player_20 <- read.csv('../fifa2020/players_20.csv', sep=",")
Teams_and_leagues <- read.csv('../fifa2020/teams_and_leagues.csv', sep=",")



Players  <- rbind( Player_15,Player_16,Player_17,Player_18,Player_19,Player_20)
View(Players)
View(Teams_and_leagues)
```

Data Cleaning

Se empezó con la limpieza de los datos verificando primero los duplicados y eliminándolos, para esto se utilizo el campo “sofifa_id” que es el identificador único de los jugadores.

```
##------------------DATA CLEANING --------------------
#verificar duplicados


# Eliminar las filas duplicadas basadas en la columna 'sofifa_id'
Players_sin_duplicados <- Players[!duplicated(Players$sofifa_id), ]

```

Después de obtener el Dataset limpio se utilizo para verificar que columnas son las que contienen datos “NA”

```
Players_NA <- colSums(is.na(Players_sin_duplicados))
Teams_and_leagues_NA <- colSums(is.na(Teams_and_leagues))


View(Players_NA)
View(Teams_and_leagues_NA)
```

En el cual se obtuvieron los siguientes resultados

Name	Type	Value
real_face	double [1]	0
release_clause_eur	double [1]	25472
player_tags	double [1]	0
team_position	double [1]	0
team_jersey_number	double [1]	669
loaned_from	double [1]	0
joined	double [1]	0
contract_valid_until	double [1]	676
nation_position	double [1]	0
nation_jersey_number	double [1]	34921
pace	double [1]	4076
shooting	double [1]	4076
passing	double [1]	4076
dribbling	double [1]	4076
defending	double [1]	4076
physic	double [1]	4076
gk_diving	double [1]	32483
gk_handling	double [1]	32483
gk_kicking	double [1]	32483
gk_reflexes	double [1]	32483
gk_speed	double [1]	32483
gk_positioning	double [1]	32483
player_traits	double [1]	0

Tomando en cuenta las columnas que se tenían con datos “NA”, se tomaron diferentes decisiones :

Para las siguientes columnas se realizo un promedio de la columna y se coloco el valor sustituyéndolo por el “NA”, esto debido a que son clausulas y habilidades.

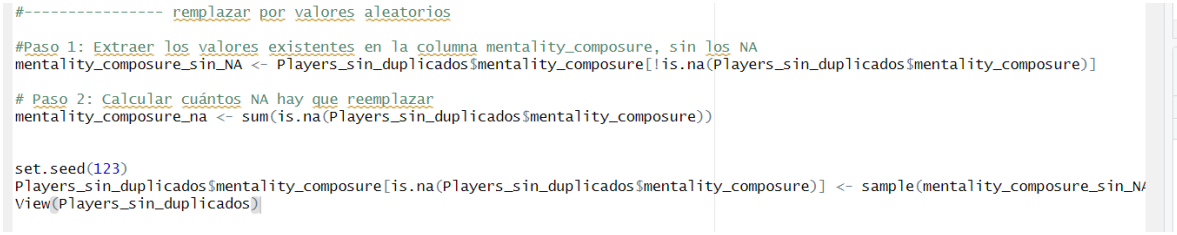
- Pace
- release_clause_eur
- passing
- dribbling
- defending
- physic

Código en R:

```r
#-------------- Limpieza de datos

# para los datos comunes se les asignara un promedio

# Calcular el promedio de la columna sin los NA de release_clause_eur
promedio_release_clause_eur <- mean(Players_sin_duplicados[["release_clause_eur"]], na.rm = TRUE)
Players_sin_duplicados[["release_clause_eur"]][is.na(Players_sin_duplicados[["release_clause_eur"]])] <- promedio_release_clause_e

# Calcular el promedio de la columna sin los NA de pace
promedio_pace <- mean(Players_sin_duplicados[["pace"]], na.rm = TRUE)
Players_sin_duplicados[["pace"]][is.na(Players_sin_duplicados[["pace"]])] <- promedio_pace

# Calcular el promedio de la columna sin los NA de shooting
promedio_shooting <- mean(Players_sin_duplicados[["shooting"]], na.rm = TRUE)
Players_sin_duplicados[["shooting"]][is.na(Players_sin_duplicados[["shooting"]])] <- promedio_shooting

# Calcular el promedio de la columna sin los NA de passing
promedio_passing <- mean(Players_sin_duplicados[["passing"]], na.rm = TRUE)
Players_sin_duplicados[["passing"]][is.na(Players_sin_duplicados[["passing"]])] <- promedio_passing

# Calcular el promedio de la columna sin los NA de dribbling
promedio_dribbling <- mean(Players_sin_duplicados[["dribbling"]], na.rm = TRUE)
Players_sin_duplicados[["dribbling"]][is.na(Players_sin_duplicados[["dribbling"]])] <- promedio_dribbling

# Calcular el promedio de la columna sin los NA de defending
promedio_defending <- mean(Players_sin_duplicados[["defending"]], na.rm = TRUE)
Players_sin_duplicados[["defending"]][is.na(Players_sin_duplicados[["defending"]])] <- promedio_defending

# Calcular el promedio de la columna sin los NA de physic
promedio_physic <- mean(Players_sin_duplicados[["physic"]], na.rm = TRUE)
Players_sin_duplicados[["physic"]][is.na(Players_sin_duplicados[["physic"]])] <- promedio_physic
```

Para la coluna “mentality_composure”, se tomo todos los datos que nos “NA” de esa columna y se agrego de forma aleatoria, ya que estos datos tenían dos formas de erce una con un número “48” y otro con un numero seguido de un signo y luego de otro número “65-2” esto indica que el jugador puede estar dentro de un rango.

Código en R:



```r
#---------------- remplazar por valores aleatorios

#Paso 1: Extraer los valores existentes en la columna mentality_composure, sin los NA
mentality_composure_sin_NA <- Players_sin_duplicados$mentality_composure[!is.na(Players_sin_duplicados$mentality_composure)]

# Paso 2: Calcular cuántos NA hay que reemplazar
mentality_composure_na <- sum(is.na(Players_sin_duplicados$mentality_composure))


set.seed(123)
Players_sin_duplicados$mentality_composure[is.na(Players_sin_duplicados$mentality_composure)] <- sample(mentality_composure_sin_NA
View(Players_sin_duplicados)
```

Para los siguientes datos se tuvo que identificar que jugadores juegan en posición de guardameta ya que las siguientes columnas son habilidades exclusivas de los porteros en la cual se incluyó aquellos jugadores que juegan en dos posiciones (incluyendo guardameta) y se agregó el promedio sustituyendo el valor “NA”, para aquellos que no juegan en la posición de guardameta se les agrego “0”.

- gk_diving
- gk_handling
- gk_kicking
- gk_reflexes
- gk_speed
- gk_positioning

Código en R:

```r
#-------------------------promedio para datos de portero unicamente

# Calcular el promedio de la columna gk_diving sin los NA
promedio_gk_diving <- mean(Players_sin_duplicados[["gk_diving"]], na.rm = TRUE)

Players_sin_duplicados[["gk_diving"]] <- sapply(1:nrow(Players_sin_duplicados), function(i) {
  if (is.na(Players_sin_duplicados[i, "gk_diving"])) {
    # Si el jugador es GK, se reemplaza con el promedio
    if (grepl("GK", Players_sin_duplicados[i, "player_positions"], ignore.case = TRUE)) {
      return(promedio_gk_diving)
    } else {
      # Si no es GK, se reemplaza con 0
      return(0)
    }
  } else {
    # Si no es NA, se deja el valor original
    return(Players_sin_duplicados[i, "gk_diving"])
  }
})



# ----- Calcular el promedio de la columna gk_handling sin los NA
promedio_gk_handling <- mean(Players_sin_duplicados[["gk_handling"]], na.rm = TRUE)

Players_sin_duplicados[["gk_handling"]] <- sapply(1:nrow(Players_sin_duplicados), function(i) {
  if (is.na(Players_sin_duplicados[i, "gk_handling"])) {
    # Si el jugador es GK, se reemplaza con el promedio
    if (grepl("GK", Players_sin_duplicados[i, "player_positions"], ignore.case = TRUE)) {
      return(promedio_gk_handling)
```

Para la parte de años de validación de contrado “contract_valid_until”, aquellos campos con “NA” se agrego “9999” para indicar año indefinido, es decir, que no tiene año de expiración de contrato.

Código en R:

```r
#--------- en contrados se coloca 9999 como indefinido ya que pueda ser que el futbolista tenga contrato indefinido
Players_sin_duplicados$contract_valid_until[is.na(Players_sin_duplicados$contract_valid_until)] <- 9999
```

Para las siguientes dos columnas que contiene “NA” al ser número de camisolas se tomo la decisión de que los números de camisolas no se pueden repetir por jugadores del mismo club es decir no existen un jugador al mismo tiempo con la misma camisola. El mismo principio se tomo para el número de camisola de su selección. Para esto se tomo los jugadores que si tuvieran número y que pertenecieran al club y se colocó un rango del 1 al 99 para agregar números de camisola y se excluye el número que ya se está utilizando por los jugadores del rango establecido.

- team_jersey_number
- nation_jersey_number

Código en R:

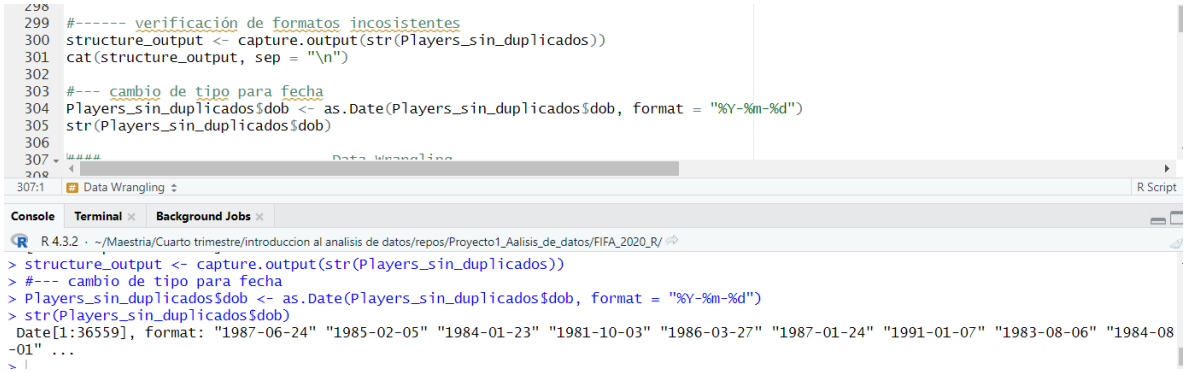
```
#------------------------------poner numero de camisola que no se repita

Players_sin_duplicados <- Players_sin_duplicados %>%
  group_by(club) %>%
  mutate(

    jersey_numbers_NA = list(team_jersey_number[!is.na(team_jersey_number)])
  ) %>%
  ungroup()


Players_sin_duplicados <- Players_sin_duplicados %>%
  group_by(club) %>%
  mutate(
    #
    jersey_numbers_disponibles = ifelse(
      length(jersey_numbers_NA[[1]]) == 0,
      list(1:99),
      list(setdiff(1:99, jersey_numbers_NA[[1]]))
    )
  ) %>%
  ungroup()

# Rellenar los valores NA en team_jersey_number con los números disponibles
Players_sin_duplicados <- Players_sin_duplicados %>%
  group_by(club) %>%
  mutate(
    team_jersey_number = ifelse(
      is.na(team_jersey_number),

      mapply(function(available_numbers, na_count) {
```

Se verifica el formato de los datos

```
 $ player_url               : chr [1:36559] "https://sofifa.com/player/158023/lionel-messi/15/157759" "https://sofifa.com/player/20801/
c-ronaldo-dos-santos-aveiro/15/157759" "https://sofifa.com/player/9014/arjen-robben/15/157759" "https://sofifa.com/player/41236/zlatan-i
brahimovic/15/157759" ...
 $ short_name               : chr [1:36559] "L. Messi" "Cristiano Ronaldo" "A. Robben" "Z. Ibrahimović" ...
 $ long_name                : chr [1:36559] "Lionel Andrés Messi Cuccittini" "Cristiano Ronaldo dos Santos Aveiro" "Arjen Robben" "Zlat
an Ibrahimović" ...
 $ age                      : int [1:36559] 27 29 30 32 28 27 23 30 29 31 ...
 $ dob                      : chr [1:36559] "1987-06-24" "1985-02-05" "1984-01-23" "1981-10-03" ...
 $ height_cm                : int [1:36559] 169 185 180 195 193 181 173 187 183 170 ...
 $ weight_kg                : int [1:36559] 67 80 80 95 92 81 74 71 79 72 ...
 $ nationality              : chr [1:36559] "Argentina" "Portugal" "Netherlands" "Sweden" ...
 $ club                     : chr [1:36559] "FC Barcelona" "Real Madrid" "FC Bayern München" "Paris Saint-Germain" ...
 $ overall                  : int [1:36559] 93 92 90 90 90 89 88 88 88 88 ...
 $ potential                : int [1:36559] 95 92 90 90 90 91 90 88 88 88 ...
 $ value_eur                : int [1:36559] 0 0 0 0 0 0 0 0 0 0 ...
 $ wage_eur                 : int [1:36559] 0 0 0 0 0 0 0 0 0 0 ...
 $ player_positions         : chr [1:36559] "CF" "LW, LM" "RM, LM, RW" "ST" ...
 $ preferred_foot           : chr [1:36559] "Left" "Right" "Left" "Right" ...
 $ international_reputation : int [1:36559] 5 5 5 5 5 5 4 5 4 4 ...
 $ weak_foot                : int [1:36559] 3 4 2 4 4 4 4 3 3 4 ...
 $ skill_moves              : int [1:36559] 4 5 4 4 1 4 4 4 3 5 ...
 $ work_rate                : chr [1:36559] "Medium/Low" "High/Low" "High/Low" "Medium/Low" ...
 $ body_type                : chr [1:36559] "Normal" "Normal" "Normal" "Normal" ...
 $ real_face                : chr [1:36559] "Yes" "Yes" "Yes" "Yes" ...
 $ release_clause_eur       : num [1:36559] 1103963 1103963 1103963 1103963 1103963 ...
 $ player_tags              : chr [1:36559] "#Speedster, #Dribbler, #FK Specialist, #Acrobat, #Clinical Finisher, #Complete Forward" "#
Speedster, #Dribbler, #Distance Shooter, #Acrobat, #Clinical Finisher, #Complete Forward" "#Speedster, #Dribbler, #Distance Shooter, #Ac
robat" "#Poacher, #Aerial Threat, #Distance Shooter, #Acrobat, #Strength, #Clinical Finisher, #Complete Forward" ...
 $ team_position            : chr [1:36559] "CF" "LW" "SUB" "ST" ...
 $ team_jersey_number       : int [1:36559] 10 7 10 10 1 9 10 20 31 7 ...
 $ loaned_from              : chr [1:36559] "" "" "" "" ...
 $ joined                   : chr [1:36559] "2004-07-01" "2009-07-01" "2009-08-28" "2012-07-01" ...
 $ contract_valid_until     : num [1:36559] 2018 2018 2017 2016 2019 ...
 $ nation_position          : chr [1:36559] "CF" "LW" "RS" "ST" ...
 $ nation_jersey_number     : int [1:36559] 10 7 11 10 1 89 10 9 7 62 ...
 $ pace                     : num [1:36559] 93 93 93 76 67.3 ...
 $ shooting                 : num [1:36559] 89 93 86 91 49.1 ...
 $ passing                  : num [1:36559] 86 81 83 81 53 ...
 $ dribbling                : num [1:36559] 96 91 92 86 58.7 ...
 $ defending                : num [1:36559] 27 32 32 34 47.4 ...
```

Se procede a cambiar el formato de la fecha de string a date

Código en R:



Data Wrangling

Se crearon las siguientes Columnas.

Edad_2024: Esta columna se creo para obtener los años año día de hoy de los jugadores a partir de la fecha de nacimiento.

Potencial_crecimento: Esta columna se crea a partir de “potential – overall”, para obtener el potencial de crecimiento de un jugador y ver donde esta ahora y que tan lejos puede llegar.

valor_relativo : Esta columna se crea a partir de “value_eur / overall”, con esto se crea un índice que combina el valor del mercado y la calificación general par obtener una media del valor relativo.

Promedio_habilidad: : Esta columna se crea a partir de “value_eur / age”, relaciona el valor del mercado con la edad del jugador para ver que tan eficiente es un jugador en relación a su valor y edad.

valor_mercado_categoria: Se agrega una categoría al jugador dependiendo del valor del mercado en que se necuentre obedeciendo los siguiente:

value_eur < 300000 = "Low",

value_eur >= 300000 & value_eur < 10000000 = "Medium",

value_eur >= 10000000 = "High"

Se convierten las habilidades en un formato largo.

```
7
8  # - convertir ancho y largo
9  # Convertir las habilidades en un formato largo
0  Players_sin_duplicados <- Players_sin_duplicados %>%
1    pivot_longer(cols = c("shooting", "passing", "dribbling", "defending", "physic"),
2                 names_to = "habilidad",
3                 values_to = "valor")
4  Players_sin_duplicados_long <- Players_sin_duplicados_long %>%
5    select(-skill)
6
```

Data Transformation

Seleccionamos la columnas númericas y se filtra el dataset, se normalizan los datos y se usara “preProcess()” del paquete “caret” que permite aplicar técnicas de preprocesamiento como normalización. Se transforman los datos en una rango de 0 a 1.

```r
##----------Data Transformation:


# Seleccionamos solo las columnas numéricas
numeric_columns <- c("age", "height_cm", "weight_kg", "overall", "potential", "value_eur", "wage_eur",
                     "pace", "gk_diving", "gk_handling", "gk_kicking", "gk_reflexes", "gk_speed", "gk_positioning",
                     "potencial_crecimento", "valor_relativo", "Promedio_habilidad", "valor_por_edad", "valor",
                     "valor_mercado_categoria")

# Filtrar solo las columnas numéricas
Players_num <- Players_sin_duplicados %>%
  select(all_of(numeric_columns))

#Normalización: Para normalizar los datos, usaremos la función preProcess() del paquete caret, que permite aplicar técnicas de pre

# Normalización de los datos usando preProcess() de caret
preprocess_model <- preProcess(Players_num, method = c("range"))

# Aplicamos la normalización al dataset
Players_normalized <- predict(preprocess_model, Players_num)
```

Datos normalizados

age	height_cm	weight_kg	overall	potential	value_eur	wage_eur	pace	gk_diving	gk_handling	gk_kicking	gk_reflexes	gk_speed
0.17857143	0.3333333	0.3934426	0.7924528	0.9090909	1.0000000	1.000	0.6266667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.17857143	0.3333333	0.3934426	0.7924528	0.9090909	1.0000000	1.000	0.6266667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.17857143	0.3333333	0.3934426	0.7924528	0.9090909	1.0000000	1.000	0.6266667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.17857143	0.3333333	0.3934426	0.7924528	0.9090909	1.0000000	1.000	0.6266667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.17857143	0.3333333	0.3934426	0.7924528	0.9090909	1.0000000	1.000	0.6266667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.14285714	0.4117647	0.3934426	0.7735849	0.9454545	0.9538462	0.920	0.8666667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
0.14285714	0.4117647	0.3934426	0.7735849	0.9454545	0.9538462	0.920	0.8666667	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

Interpretación de las columnas normalizadas

1. **age (Edad):**

    - Esto indica que los valores de edad han sido escalados a este rango (0 y 1). Si por ejemplo la edad original de un jugador fuera de 20 años y el rango de edad original fuera de 18 a 40 años, 20 años sería un valor cercano a 0.393.

2. **height_cm (Altura en cm):**

    - Los valores de altura (por ejemplo, 0.294 y 0.608) se normalizan entre 0 y 1. Esto sugiere que la altura de los jugadores se encuentra en un rango determinado, y la normalización ajusta esos valores al nuevo rango. Un valor de 0.294 es cercano al mínimo, y 0.608 es relativamente más alto, pero aún por debajo del máximo.

3. **weight_kg (Peso en kg):**

    - Similar a la altura, los valores de peso también están normalizados. Los valores 0.295 a 0.508 indican que el peso de los jugadores en el conjunto de datos se encuentra dentro de un rango determinado, con los valores más cercanos a 0 indicando los jugadores más ligeros y los más cercanos a 1 los más pesados.

4. **overall (Valoración general):**

    - La columna overall ha sido normalizada entre 0 y 1, con los jugadores más valorados en la parte superior del rango. El valor 1 en la primera fila indica que este jugador tiene la valoración máxima posible dentro de este dataset. Un valor de 0.981 indica un jugador que está cerca del máximo, pero no lo alcanza.

5. **potential (Potencial):**

    - La columna de potencial también muestra valores cercanos a 1 para los jugadores con mayor potencial. Un valor de 0.945 significa que este jugador tiene un potencial muy alto, pero no el máximo.

6. **value_eur (Valor en euros):**

    - Los valores de los jugadores en euros han sido normalizados entre 0 y 1. Un valor de 0 indica que el jugador no tiene valor o tiene un valor muy bajo, mientras que el valor 0 sugiere que estos jugadores tienen un valor de mercado muy.

7. **wage_eur (Salario en euros):**

    - Los salarios también están normalizados entre 0 y 1 y dependiendo del salario se puede categorizar al jugador.

8. **pace, gk_diving, gk_handling, gk_kicking, gk_reflexes, gk_speed (Y otras habilidades específicas de los jugadores):**
    - Estas columnas representan habilidades o atributos específicos de los jugadores. Los valores de estas columnas son valores normalizados entre 0 y 1, lo que significa que los jugadores que sobresalen en estas habilidades tendrán valores cercanos a 1, mientras que aquellos con menor habilidad tendrán valores cercanos a 0.

Estandarización

```
preprocess_model_std <- preProcess(Players_num, method = c("center", "scale"))

# Aplicar la estandarización al dataset
Players_standardized <- predict(preprocess_model_std, Players_num)

# Ver los primeros resultados
head(Players_standardized)
```

age	height_cm	weight_kg	overall	potential	value_eur	wage_eur	pace	gk_diving	gk_handling	gk_kicking	gk_reflexes	gk_speed
0.7632645	-1.78435671	-1.11776951	4.595211	4.209586	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994
0.7632645	-1.78435671	-1.11776951	4.595211	4.209586	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994
0.7632645	-1.78435671	-1.11776951	4.595211	4.209586	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994
0.7632645	-1.78435671	-1.11776951	4.595211	4.209586	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994
0.7632645	-1.78435671	-1.11776951	4.595211	4.209586	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994
1.1961459	0.62392375	0.77412735	4.446662	3.728592	-0.3731126	-0.3862503	2.7093332	-0.3513501	-0.3509535	-0.3507456	-0.3510084	-0.3392994

Interpretación de columnas estandarizadas:

1. **age (Edad)**:
    - El valor **0.763** en las primeras filas indica que estos jugadores tienen una edad superior al promedio de la muestra. Un valor de **1.20** en la fila 6 sugiere que este jugador es significativamente mayor que la media de la muestra en términos de edad.
2. **height_cm (Altura en cm)**:
    - Los valores negativos (por ejemplo, **-1.78**) indican que estos jugadores tienen una altura por debajo de la media de la muestra. Esto significa que la mayoría de los jugadores en el conjunto de datos tienen una altura superior a los valores de estas primeras filas.

- Un valor positivo (**0.624**) indica que este jugador es más alto que la media del conjunto de datos.

3. **weight_kg (Peso en kg)**:
   - Al igual que con la altura, los valores negativos (como **-1.12**) indican que estos jugadores tienen un peso inferior a la media de la muestra. El valor positivo de **0.774** indica que el jugador de la fila 6 tiene un peso superior a la media.

4. **overall (Valoración general)**:
   - Un valor de **4.60** en varias filas indica que estos jugadores tienen una valoración general bastante alta, por encima de la media del conjunto de datos (suponiendo que el valor máximo sea superior a 4.60).
   - El jugador de la fila 6 tiene una valoración de **4.45**, que sigue siendo alta, pero ligeramente por debajo de los primeros jugadores.
5. **potential (Potencial)**:
   - Los jugadores con un **potencial de 4.21** en varias filas son valorados como teniendo un potencial alto (por encima de la media).
   - El jugador de la fila 6, con un valor de **3.73**, tiene un potencial ligeramente inferior al de los demás.
6. **value_eur (Valor en euros)** y **wage_eur (Salario en euros)**:
   - Los valores **-0.373** y **-0.386** indican que estos jugadores tienen valores por debajo de la media en cuanto a su valor de mercado y salario. Los valores negativos generalmente implican que estos jugadores tienen un valor de mercado o salario más bajo en comparación con el resto de la muestra.
7. **pace (Velocidad)**:
   - El valor **2.71** en todas las filas indica que la velocidad de estos jugadores es mucho más alta que la media del conjunto de datos (2.71 es un valor bastante alto en una distribución normalizada).
8. **gk_diving, gk_handling, gk_kicking, gk_reflexes, gk_speed (Habilidades de portero):**
   - Los valores de **-0.351** en todas estas habilidades indican que estos jugadores tienen habilidades inferiores al promedio para estas

características específicas. Este patrón sugiere que estos jugadores no son porteros de alto nivel, según sus habilidades estandarizadas.

Datos estadísticos

Datos originales:

Variable	Mínimo	1er Cuartil	Mediana	Media	3er Cuartil	Máximo
age	16.00	20.00	23.00	23.47	27.00	44.00
height_cm	154.0	176.0	181.0	180.9	185.0	205.0
weight_kg	49.00	70.00	75.00	74.68	79.00	110.00
overall	40.00	57.00	62.00	62.07	66.00	93.00
potential	40.00	65.00	68.00	68.74	73.00	95.00
value_eur	0	0	90,000	325,908	350,000	32,500,000
wage_eur	0	0	1,000	1,685	2,000	125,000
pace	22.00	62.00	67.31	67.31	73.00	97.00
gk_diving	0	0	0	6.91	0	88.00
gk_handling	0	0	0	6.58	0	87.00
gk_kicking	0	0	0	6.47	0	92.00
gk_reflexes	0	0	0	7.01	0	90.00
gk_speed	0	0	0	4.30	0	68.00

Datos normalizados:

Variable	Mínimo	1er Cuartil	Mediana	Media	3er Cuartil	Máximo
age	0.0000	0.1429	0.2500	0.2669	0.3929	1.0000
height_cm	0.0000	0.4314	0.5294	0.5266	0.6078	1.0000
weight_kg	0.0000	0.3443	0.4262	0.4210	0.4918	1.0000
overall	0.0000	0.3208	0.4151	0.4163	0.4906	1.0000
potential	0.0000	0.4545	0.5091	0.5226	0.6000	1.0000
value_eur	0.0000	0.0000	0.0028	0.0100	0.0108	1.0000
wage_eur	0.0000	0.0000	0.0080	0.0135	0.0160	1.0000
pace	0.0000	0.5333	0.6042	0.6042	0.6800	1.0000
gk_diving	0.0000	0.0000	0.0000	0.0785	0.0000	1.0000
gk_handling	0.0000	0.0000	0.0000	0.0757	0.0000	1.0000
gk_kicking	0.0000	0.0000	0.0000	0.0703	0.0000	1.0000
gk_reflexes	0.0000	0.0000	0.0000	0.0778	0.0000	1.0000
gk_speed	0.0000	0.0000	0.0000	0.0632	0.0000	1.0000

Datos estandarizados:

Variable	Mínimo	1er Cuartil	Mediana	Media	3er Cuartil	Máximo
age	-1.6176	-0.7518	-0.1025	0.0000	0.7633	4.4428
height_cm	-4.0421	-0.7307	0.0219	0.0000	0.6239	3.6343
weight_kg	-3.7373	-0.6812	0.0465	0.0000	0.6286	5.1400
overall	-3.2779	-0.7526	-0.0098	0.0000	0.5844	4.5952
potential	-4.6086	-0.6004	-0.1194	0.0000	0.6823	4.2096
value_eur	-0.3731	-0.3731	-0.2701	0.0000	0.0276	36.8342
wage_eur	-0.3863	-0.3863	-0.1570	0.0000	0.0723	28.2706
pace	-4.7794	-0.5604	0.0000	0.0000	0.5998	3.1312
gk_diving	-0.3513	-0.3513	-0.3513	0.0000	-0.3513	4.1263
gk_handling	-0.3510	-0.3510	-0.3510	0.0000	-0.3510	4.2870
gk_kicking	-0.3507	-0.3507	-0.3507	0.0000	-0.3507	4.6385
gk_reflexes	-0.3510	-0.3510	-0.3510	0.0000	-0.3510	4.1580
gk_speed	-0.3393	-0.3393	-0.3393	0.0000	-0.3393	5.0310

Conclusiones

- La edad de los jugadores oscila entre los 16 y 44 años, con una media de aproximadamente 23.5 años. Esto sugiere que la mayoría de los jugadores en los datos son relativamente jóvenes, con un fuerte enfoque en el talento emergente.

- La altura promedio es de 180.9 cm, con un rango que va desde 154 cm hasta 205 cm, lo que cubre una gran variedad de tipos físicos. El peso promedio es de 74.68 kg, lo que refleja una amplia gama de físicos, desde jugadores más ligeros hasta los más corpulentos.

- El valor promedio de los jugadores es de 325,908 EUR, mientras que el salario promedio es de 1,685 EUR. Sin embargo, el rango es extenso, con valores que varían desde 0 EUR hasta 32.5 millones de EUR, lo que indica la presencia de jugadores de élite que dominan las métricas.

- La habilidad promedio (overall) de los jugadores es de 62.07, lo que es consistente con una población de jugadores profesionales que incluyen tanto talentos emergentes como jugadores ya consolidados. La habilidad de los porteros (representada por las métricas de "gk") muestra valores bajos, ya que solo algunos jugadores son especializados en este rol.

- Al comparar los datos normalizados y estandarizados, se observa que ambos procesos afectan las métricas. El rango de valores en los datos normalizados va de 0 a 1, mientras que en los estandarizados va de valores negativos a positivos, con la media acercándose a 0. Esto hace que los datos estandarizados sean útiles para modelos de machine learning que asuman distribuciones estándar, mientras que los datos normalizados son más intuitivos para interpretar las proporciones relativas entre los jugadores.