Informe limpieza de datos y transformacion

Estudiante:

Isabella Sofia Alvis Caballero

Semestre:

Sexto semestre

Cartagena de Indias

Fundacion Universitaria Tecnologico Comfenalco

- **Descripción de la fuente de datos empleada.**

  La fuente de datos empleada proviene de un archivo CSV titulado **all_players.csv**, el cual se encuentra alojado en un repositorio público de GitHub. Este archivo contiene datos relacionados con jugadores de fútbol, sus habilidades y características. La información clave incluye atributos como velocidad de sprint, precisión de pase, remates, control del balón, y diversas métricas físicas y técnicas.

- Para este análisis de datos, utilizamos varios paquetes de Python esenciales:

Importamos los paquetes que vamos a usar

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import  SimpleImputer
```

- Se busca un dataset el cual debe contiene 17.000 registros. Los datos se cargaron desde un archivo CSV en GitHub con éxito. La advertencia indica que hay columnas con tipos de datos mixtos, lo que puede requerir un tratamiento especial.

Cargamos los datos

Follow link (ctrl + click)

Loading...

```
url = 'https://raw.githubusercontent.com/LalaA29/Actividad-limpieza-datos/refs/heads/main/all_players.csv'
data=pd.read_csv(url)
print("Datos desde la URL cargados con éxito.")
```

```
Datos desde la URL cargados con éxito.
<ipython-input-2-3cf913d6aafd>:2: DtypeWarning: Columns (33) have mixed types. Specify dtype option on impor
  data=pd.read_csv(url)
```

- Este comando nos muestra los datos de la tabla

  Head: Muestra las primeras cinco filas del conjunto de datos.

```
data.head()
```

1 to 5 of 5 entries Filter

index	Unnamed: 0	Rank	Name	OVR	PAC	SHO	PAS	DRI	DEF	PHY	Acceleration	Sprint Speed	Positioning	Finishin
2	2	4	Erling Haaland	91	88	92	70	81	45	88	80	94	96	96
0	0	1	Kylian Mbappé	91	97	90	80	92	36	78	97	97	93	94
3	3	5	Jude Bellingham	90	80	87	83	88	78	83	81	80	91	90
4	4	7	Vini Jr.	90	95	84	81	91	29	69	95	95	87	89
1	1	2	Rodri	91	66	80	86	84	87	85	65	66	76	74

Show 25 per page

La estructura inicial del dataset incluye 1453 filas y 17 columnas, con datos de diferentes tipos (object, float64, int64).

Info: Proporciona el número de entradas, columnas y tipos de datos.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1453 entries, 0 to 1452
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   state                      1453 non-null   object
 1   state_code                 1453 non-null   object
 2   num_colonies               1453 non-null   int64
 3   max_colonies               1453 non-null   int64
 4   lost_colonies              1453 non-null   int64
 5   percent_lost               1453 non-null   int64
 6   added_colonies             1453 non-null   int64
 7   renovated_colonies         1453 non-null   int64
 8   percent_renovated          1453 non-null   int64
 9   quarter                    1453 non-null   int64
 10  year                       1453 non-null   int64
 11  varroa_mites               1453 non-null   float64
 12  other_pests_and_parasites  1453 non-null   float64
 13  diseases                   1453 non-null   float64
 14  pesticides                 1453 non-null   float64
 15  other                      1453 non-null   float64
 16  unknown                    1453 non-null   float64
dtypes: float64(6), int64(9), object(2)
memory usage: 193.1+ KB
```

Para cambiar el tipo de dato de una columna

- Se modifica el tipo de dato de un atributo

La columna 'Sprint Speed' tenía valores almacenados como texto, pero se corrigió a tipo numérico para su análisis.

```
data['Sprint Speed']=data['Sprint Speed'].astype(str)
data.info()
```

```
 1   Rank                 17737 non-null  int64
 2   Name                 17737 non-null  object
 3   OVR                  17737 non-null  int64
 4   PAC                  17737 non-null  int64
 5   SHO                  17737 non-null  int64
 6   PAS                  17737 non-null  int64
 7   DRI                  17737 non-null  int64
 8   DEF                  17737 non-null  int64
 9   PHY                  17737 non-null  int64
 10  Acceleration         17737 non-null  int64
 11  Sprint Speed         17737 non-null  object
 12  Positioning          17737 non-null  object
 13  Finishing            17737 non-null  object
 14  Shot Power           17737 non-null  int64
 15  Long Shots           17737 non-null  object
 16  Volleys              17737 non-null  object
 17  Penalties            17737 non-null  int64
 18  Vision               17737 non-null  int64
 19  Crossing             17737 non-null  object
 20  Free Kick Accuracy   17737 non-null  object
 21  Short Passing        17737 non-null  int64
 22  Long Passing         17737 non-null  int64
 23  Curve                17737 non-null  object
 24  Dribbling            17737 non-null  object
 25  Agility              17737 non-null  int64
 26  Balance              17737 non-null  int64
 27  Reactions            17737 non-null  int64
```

Descripción estadística de los datos.

Se generaron descripciones estadísticas para las columnas numéricas y categóricas, lo que permite una vista rápida de métricas como la media, mediana, y desviación estándar para las variables numéricas.

```
data['Sprint Speed']=data['Sprint Speed'].astype(int)
```

Para obtener una descrpcion estadistica de las columnas numericas

```
[8] data.describe()
```

	Unnamed: 0	Rank	OVR	PAC	SHO	PAS	DRI	
count	17737.000000	17737.000000	17737.000000	17737.000000	17737.000000	17737.000000	17737.000000	17737.
mean	8868.000000	8503.317021	66.796245	68.237921	54.846197	59.152506	64.205728	51.
std	5120.375198	5109.915335	7.030997	10.587756	13.858698	9.826691	9.359164	16.
min	0.000000	1.000000	47.000000	29.000000	20.000000	25.000000	29.000000	15.
25%	4434.000000	4260.000000	62.000000	62.000000	46.000000	53.000000	59.000000	37.
50%	8868.000000	7937.000000	67.000000	69.000000	57.000000	60.000000	65.000000	55.
75%	13302.000000	13135.000000	71.000000	75.000000	65.000000	66.000000	70.000000	64.
max	17736.000000	17717.000000	91.000000	97.000000	93.000000	94.000000	92.000000	91.

8 rows × 33 columns

- Para columnas categóricas

Para obtener una descripcion de las categoricas

```
[9] data.describe(include=['O'])
```

	Name	Positioning	Finishing	Long Shots	Volleys	Crossing	Free Kick Accuracy	Curve	Dribbling	Interceptions	.
count	17737	17737	17737	17737	17737	17737	17737	17737	17737	17737	
unique	17608	93	94	86	88	87	87	87	90	85	
top	Danilo	58	60	60	48	60	42	58	65	64	
freq	4	570	500	522	431	593	491	460	734	553	

4 rows × 24 columns

- Devolverá una representación las dimensiones del arreglo (número de filas, número de columnas).

Para ver el tamaño del dataset

```
data.shape
```

```
(17737, 57)
```

- **Detección de Outliers**

  La detección de outliers se realizó visual y numéricamente utilizando el rango intercuartílico (IQR) y desviación estándar:

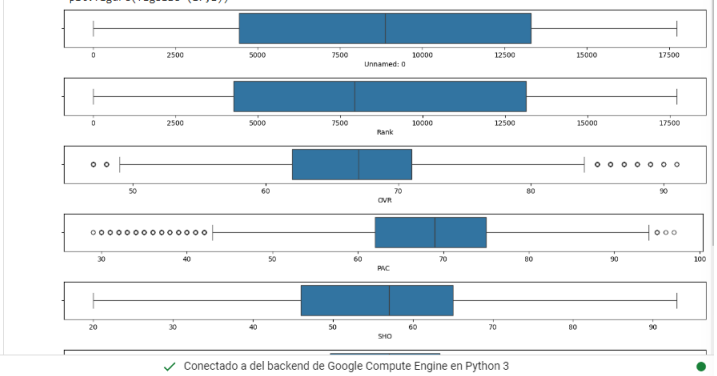
COMPROBACION DE OUTLIERS

De forma visual

```
[11] num_cols=data.select_dtypes(include='number').columns

     for col in num_cols:
       plt.figure(figsize=(17,1))
       sns.boxplot(data=data[num_cols], x=col)
```

```
<ipython-input-11-77c43ab28d5e>:4: RuntimeWarning: More than 20 figures have been opened. Figures created
  plt.figure(figsize=(17,1))
```



Conectado a del backend de Google Compute Engine en Python 3

```python
#Nº de Outliers usando rango IQR
num_cols=data.select_dtypes(include='number').columns
outliers={}

for col in num_cols:
  Q1 = np.percentile(data[col], 25)
  Q3 = np.percentile(data[col], 75)
  IQR = Q3 - Q1
  lower_bound = Q1 - 1.5 * IQR
  upper_bound = Q3 + 1.5 * IQR
  outliers[col]= (data[col] > upper_bound).sum() + (data[col]< lower_bound).sum()

print(outliers)
```

- **Parámetros relevantes utilizados en los diferentes algoritmos.**
- plt.figure(figsize=(17,1))

Este es un comando de matplotlib, una biblioteca en Python que se usa para crear gráficos. Aquí, se está creando una nueva figura, que es el lienzo donde se dibujará el gráfico. El argumento figsize=(17,1) establece el tamaño de la figura.

figsize=(17,1): Define el tamaño de la figura en pulgadas (ancho y alto). En este caso, se crea una figura bastante ancha (17 unidades) pero muy baja (1 unidad), lo que resulta útil para representar gráficos largos y delgados, como boxplots horizontales.

- sns.boxplot(data=data[num_cols], x=col)

Este es un comando de seaborn (importado como sns), que es una biblioteca de visualización de datos basada en matplotlib y que proporciona gráficos estadísticos más atractivos y fáciles de interpretar.

- **Q1 = np.percentile(data[col], 25)**

Aquí se está calculando el **primer cuartil** (Q1) de los datos en la columna col del DataFrame data. El cuartil es un valor que divide los datos en cuatro partes iguales, y el **primer cuartil (Q1)** es el valor debajo del cual se encuentra el 25% de los datos.

np.percentile(data[col], 25): La función np.percentile de la biblioteca **NumPy** se usa para calcular percentiles. En este caso, se está calculando el percentil 25, que corresponde a Q1.

- Q3 = np.percentile(data[col], 75)

Similar al cálculo anterior, esta línea calcula el **tercer cuartil (Q3)**, que es el valor debajo del cual se encuentra el 75% de los datos. Es decir, el 25% de los datos son mayores que este valor.

np.percentile(data[col], 75): Calcula el percentil 75, que corresponde a Q3.

- IQR = Q3 - Q1

Aquí se está calculando el **rango intercuartílico (IQR)**. El IQR es la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1), y mide la dispersión en la parte central de los datos.

IQR: Es una medida de variabilidad que elimina los efectos de los valores extremos, ya que se basa en la diferencia entre los cuartiles.

- **lower_bound = Q1 - 1.5 * IQR**

Esta línea calcula el **límite inferior** para identificar valores atípicos. Se basa en la regla del rango intercuartílico (IQR), que establece que cualquier dato por debajo de 1.5 * IQR del primer cuartil se considera un outlier.

lower_bound: Es el límite inferior, más allá del cual los valores se consideran atípicos por ser extremadamente bajos.

- **upper_bound = Q3 + 1.5 * IQR**

De forma similar al cálculo anterior, esta línea calcula el **límite superior** para los valores atípicos. Cualquier dato que esté por encima de 1.5 * IQR del tercer cuartil es considerado un outlier.

upper_bound: Es el límite superior, más allá del cual los valores se consideran atípicos por ser extremadamente altos.

- **outliers[col]= (data[col] > upper_bound).sum() + (data[col] < lower_bound).sum()**

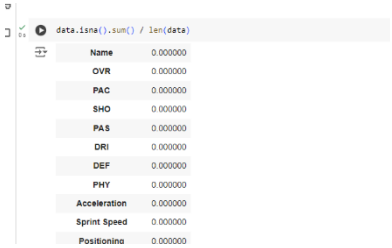
Esta línea cuenta cuántos valores en la columna col del DataFrame data son mayores que el límite superior (upper_bound) o menores que el límite inferior (lower_bound), y luego los suma. El resultado se almacena en un diccionario o DataFrame llamado outliers con la clave col, que representa el número total de outliers en esa columna.

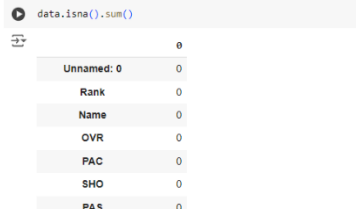
- Manejo de Valores Nulos

Revisión de Nulos: Se verificó la cantidad de valores nulos en cada columna.

Porcentaje de Nulos: Se calculó el porcentaje de datos faltantes por columna.

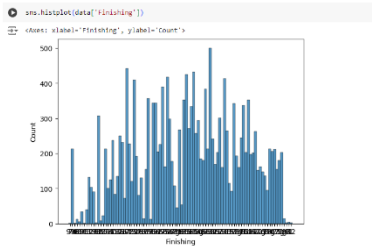
Eliminación de Nulos: Se eliminó la columna 'Sprint Speed' debido a la gran cantidad de valores nulos.





Imputación de Valores

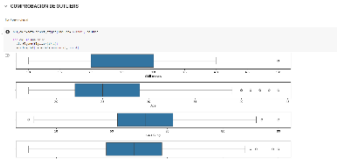
Se visualizó la distribución de algunas columnas antes de proceder con la imputación de valores faltantes. Para la imputación, se podrían utilizar estrategias como la media, la mediana o un valor constante.



- **Resultados obtenidos por los diferentes códigos escogidos de forma gráfica y comparada/superpuesta.**

Gráfico de Boxplots para Outliers

Este gráfico muestra la distribución de valores numéricos y resalta los outliers. Se superponen las diferentes columnas numéricas del dataset.



Distribución de Datos (Histograma)

El histograma permite visualizar la distribución de los datos para una columna específica, útil para detectar sesgos o valores atípicos.

Comparación de Outliers (IQR vs 3 std)

Para comparar los resultados obtenidos con diferentes métodos de detección de outliers (rango intercuartil - IQR y desviación estándar - std), podemos superponer o comparar los resultados numéricamente.

Gráfico de Datos Faltantes

Un gráfico de barras puede visualizar qué columnas tienen valores nulos y en qué porcentaje.

- **Discusión de los resultados obtenidos y argumentos sobre cómo mejorar de dichos resultados.**