Reporte Limpieza de Datos.

David Lopez.
Introducción a la Ciencia de Datos.
Jaime Alejandro Romero Sierra.

Link Github: https://github.com/davidLOPEZ24/proyectoCdeD

Fuente o contexto de la base de datos

La base de datos utilizada en este proyecto fue construida a partir de información recopilada sobre el rendimiento de jugadores en la **National Basketball Association (NBA)**. Esta liga es considerada la máxima competencia de baloncesto profesional en el mundo y cuenta con un amplio registro estadístico que permite analizar el desempeño de los atletas en múltiples dimensiones.

El dataset reúne información de distintas temporadas, en este caso correspondiente al periodo **1997–1998**, e incluye variables que describen tanto características personales de los jugadores (como nombre, edad, posición y equipo) como métricas detalladas de su rendimiento en cancha: minutos jugados, porcentaje de tiro, rebotes, asistencias, robos, bloqueos y puntos, entre otras.

La fuente de los datos proviene de sitios especializados en estadísticas deportivas como **Basketball Reference**, que recopilan y publican información oficial de cada temporada de la NBA. Posteriormente, estos registros fueron organizados, limpiados y estructurados con el objetivo de crear una base de datos coherente y lista para su análisis.

El propósito de trabajar con esta información fue **explorar patrones de rendimiento, realizar procesos de limpieza y transformación de datos, y aplicar técnicas de análisis estadístico y visualización** dentro del contexto deportivo. Además, se eligió esta temática porque combina dos áreas de gran interés: el deporte y la ciencia de datos, permitiendo demostrar cómo la analítica puede aportar valor en la evaluación del desempeño atlético y la toma de decisiones estratégicas en el baloncesto profesional.

Descripción general del contenido

La base de datos contiene información detallada sobre los jugadores que participaron en la temporada **1997–1998 de la National Basketball Association (NBA)**. Cada registro corresponde a un jugador único en esa temporada, e incluye tanto sus datos personales como estadísticas obtenidas durante los partidos oficiales disputados.

Las principales categorías de información son:

● **Datos personales y de identificación:** nombre del jugador, posición en el campo, edad y equipo al que perteneció durante la temporada.

● **Indicadores de participación:** número de juegos disputados, titularidades, y promedio de minutos jugados por partido, los cuales reflejan la constancia y el rol del jugador dentro del equipo.

● **Rendimiento ofensivo:** estadísticas relacionadas con la anotación, tales como tiros de campo intentados y encestados, porcentaje de acierto, tiros de tres puntos, tiros libres y puntos totales.

● **Rendimiento defensivo y general:** métricas que evalúan la contribución del jugador en defensa y en juego colectivo, como rebotes ofensivos y defensivos, asistencias, robos, bloqueos, pérdidas y faltas personales.

En términos de estructura, la base cuenta con **31 columnas y más de 400 registros**, lo que permitió realizar un proceso de limpieza y estandarización significativo. Durante esta etapa se identificaron **valores nulos, duplicados y formatos inconsistentes**, como porcentajes o números almacenados como texto. El proceso de depuración incluyó la **corrección de tipos de datos, eliminación de duplicados y normalización de valores**, garantizando la coherencia de toda la información.

El resultado fue una base de datos limpia, homogénea y completamente funcional para su análisis posterior. Este conjunto de datos no solo permite estudiar el rendimiento individual de los jugadores, sino también **comparar posiciones, equipos o tendencias estadísticas**, abriendo la posibilidad de desarrollar proyectos de **análisis predictivo o minería de datos aplicada al baloncesto profesional**.

.

Significado de cada columna

Rango: Número de registro o posición del jugador dentro de la lista de estadísticas oficiales.
Jugador: Nombre completo del jugador.
Posición: Posición en la que juega el jugador (PG = Base, SG = Escolta, SF = Alero, PF = Ala-Pívot, C = Pívot).
Edad: Edad del jugador durante la temporada analizada.
Equipo: Equipo al que perteneció el jugador (abreviatura oficial del equipo de la NBA).
Juegos: Número total de partidos disputados por el jugador en la temporada.
Juegos_Iniciados: Número total de partidos en los que el jugador fue titular ("Games Started").
Minutos_Juego: Promedio de minutos jugados por partido.
Tiros_Campo_Anotados: Promedio de tiros de campo encestados por partido.
Tiros_Campo_Intentados: Promedio de tiros de campo intentados por partido.
Porcentaje_Tiros_Campo: Porcentaje de efectividad en tiros de campo.
Triples_Anotados: Promedio de tiros de tres puntos encestados por partido.
Triples_Intentados: Promedio de tiros de tres puntos intentados por partido.
Porcentaje_Triples: Porcentaje de efectividad en tiros de tres puntos.
Dobles_Anotados: Promedio de tiros de dos puntos encestados por partido.
Dobles_Intentados: Promedio de tiros de dos puntos intentados por partido.
Porcentaje_Dobles: Porcentaje de efectividad en tiros de dos puntos.
Porcentaje_Efectivo_Tiro_Campo: Porcentaje efectivo de tiro de campo, que ajusta por el valor adicional del triple.
Tiros_Libres_Anotados: Promedio de tiros libres encestados por partido.
Tiros_Libres_Intentados: Promedio de tiros libres intentados por partido.
Porcentaje_Tiros_Libres: Porcentaje de efectividad en tiros libres.
Rebotes_Ofensivos: Promedio de rebotes ofensivos capturados por partido.
Rebotes_Defensivos: Promedio de rebotes defensivos capturados por partido.
Rebotes_Totales: Promedio total de rebotes por partido.
Asistencias: Promedio de asistencias registradas por partido.
Robos: Promedio de robos de balón logrados por partido.
Bloqueos: Promedio de tiros bloqueados por partido.
Pérdidas: Promedio de pérdidas de balón ("turnovers") por partido.
Faltas: Promedio de faltas personales cometidas por partido.
Puntos: Promedio de puntos anotados por partido.
Temporada: Año o temporada de la NBA a la que pertenecen los datos (en este caso, 1997–1998).

Proceso de Limpieza

1. Cargar Librerías y Base de datos.

```python
# Cargar datos desde archivo CSV
import pandas as pd
import numpy as np
df=pd.read_csv('df_sucio.csv')
df
```

✓ 5.2s Python

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year
0	1	Mahmoud Abdul-Rauf	PG	28	SAC	31.0	0	17.1	3.3	8.8	...	NaN	1.0	1.2	1.9	0.5	0.0	0.6	1.0	7.3	1997-1998
1	2	NaN	SG	23	SAC	59.0	16	16.3	2.4	6.1	...	0.7	1.2	2.0	NaN	0.6	0.2	1.1	1.4	6.4	1997-1998
2	3	Shareef Abdur-Rahim	SF	21	VAN	82.0	82	36.0	8.0	16.4	...	2.8	4.3	7.1	2.6	1.1	0.9	3.1	2.5	22.3	1997-1998
3	4	Cory Alexander	PG	24	TOT	60.0	22	21.6	2.9	6.7	...	0.3	2.2	2.4	3.5	1.2	0.2	1.9	1.6	8.1	1997-1998
4	4	NaN	PG	24	SAS	37.0	3	13.5	1.6	3.9	...	0.2	1.1	1.3	1.9	0.7	0.1	1.3	1.4	4.5	1997-1998
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18470	186	Josh Howard	SF	23	DAL	67.0	29	23.7	3.4	Auto%#	...	2.2	3.3	5.5	1.4	1.0	0.8	1.0	2.5	8.6	2003-2004
18471	358	Charles Smith	SG	22	TOT	34.0	0	8.6	1.4	3.7	...	0.4	0.4	0.8	0.6	0.4	0.2	0.8	0.7	3.5	1997-1998

2. Mostrar información general del DataFrame

```python
# Mostrar información general del DataFrame
df.info()
```

✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18475 entries, 0 to 18474
Data columns (total 31 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Rk      17921 non-null  object
 1   Player  17921 non-null  object
 2   Pos     17921 non-null  object
 3   Age     17921 non-null  object
 4   Tm      17921 non-null  object
 5   G       17921 non-null  float64
 6   GS      17921 non-null  object
 7   MP      17921 non-null  float64
 8   FG      17921 non-null  float64
 9   FGA     17921 non-null  object
 10  FG%     17819 non-null  float64
 11  3P      17921 non-null  object
 12  3PA     17921 non-null  float64
 13  3P%     15237 non-null  float64
 14  2P      17921 non-null  float64
 15  2PA     17921 non-null  float64
 16  2P%     17737 non-null  object
 17  eFG%    17823 non-null  object
 18  FT      17921 non-null  object
```

3. Mostrar primeras filas del DataFrame

```
# Mostrar las primeras filas del dataset
df.head()
```

✓ 0.0s Python

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	...	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	Year
0	1	Mahmoud Abdul-Rauf	PG	28	SAC	31.0	0	17.1	3.3	8.8	...	NaN	1.0	1.2	1.9	0.5	0.0	0.6	1.0	7.3	1997-1998
1	2	NaN	SG	23	SAC	59.0	16	16.3	2.4	6.1	...	0.7	1.2	2.0	NaN	0.6	0.2	1.1	1.4	6.4	1997-1998
2	3	Shareef Abdur-Rahim	SF	21	VAN	82.0	82	36.0	8.0	16.4	...	2.8	4.3	7.1	2.6	1.1	0.9	3.1	2.5	22.3	1997-1998
3	4	Cory Alexander	PG	24	TOT	60.0	22	21.6	2.9	6.7	...	0.3	2.2	2.4	3.5	1.2	0.2	1.9	1.6	8.1	1997-1998
4	4	NaN	PG	24	SAS	37.0	3	13.5	1.6	3.9	...	0.2	1.1	1.3	1.9	0.7	0.1	1.3	1.4	4.5	1997-1998

4. Renombramos las columnas

```python
#Renombrar las columnas al español
# # Diccionario de traducción de nombres
nuevos_nombres = {
    'Rk': 'Rango', 'Player': 'Jugador', 'Pos': 'Posición', 'Age': 'Edad', 'Tm': 'Equipo',
    'G': 'Juegos', 'GS': 'Juegos_Iniciados', 'MP': 'Minutos_Juego', 'FG': 'Tiros_Campo_Anotados',
    'FGA': 'Tiros_Campo_Intentados', 'FG%': 'Porcentaje_Tiros_Campo', '3P': 'Triples_Anotados',
    '3PA': 'Triples_Intentados', '3P%': 'Porcentaje_Triples', '2P': 'Dobles_Anotados',
    '2PA': 'Dobles_Intentados', '2P%': 'Porcentaje_Dobles', 'eFG%': 'Porcentaje_Efectivo_Tiro_Campo',
    'FT': 'Tiros_Libres_Anotados', 'FTA': 'Tiros_Libres_Intentados', 'FT%': 'Porcentaje_Tiros_Libres',
    'ORB': 'Rebotes_Ofensivos', 'DRB': 'Rebotes_Defensivos', 'TRB': 'Rebotes_Totales',
    'AST': 'Asistencias', 'STL': 'Robos', 'BLK': 'Bloqueos', 'TOV': 'Pérdidas',
    'PF': 'Faltas', 'PTS': 'Puntos', 'Year': 'Temporada'
}
```

✓ 0.0s Python

5. Renombrar columnas del DataFrame

```python
# Renombrar columnas del DataFrame
df.rename(columns=nuevos_nombres, inplace=True)
```

✓ 0.0s Python

6. Ejecución o transformación de datos

```python
# Ejecución o transformación de datos
df.columns
```

✓ 0.0s Python

```
Index(['Rango', 'Jugador', 'Posición', 'Edad', 'Equipo', 'Juegos',
       'Juegos_Iniciados', 'Minutos_Juego', 'Tiros_Campo_Anotados',
       'Tiros_Campo_Intentados', 'Porcentaje_Tiros_Campo', 'Triples_Anotados',
       'Triples_Intentados', 'Porcentaje_Triples', 'Dobles_Anotados',
       'Dobles_Intentados', 'Porcentaje_Dobles',
       'Porcentaje_Efectivo_Tiro_Campo', 'Tiros_Libres_Anotados',
       'Tiros_Libres_Intentados', 'Porcentaje_Tiros_Libres',
       'Rebotes_Ofensivos', 'Rebotes_Defensivos', 'Rebotes_Totales',
       'Asistencias', 'Robos', 'Bloqueos', 'Pérdidas', 'Faltas', 'Puntos',
       'Temporada'],
      dtype='object')
```

7. Vista General

```python
# Vista general
df.isnull().sum()
```

✓ 0.0s Python

```
Rango                          554
Jugador                        554
Posición                       554
Edad                           554
Equipo                         554
Juegos                         554
Juegos_Iniciados               554
Minutos_Juego                  554
Tiros_Campo_Anotados           554
Tiros_Campo_Intentados         554
Porcentaje_Tiros_Campo         656
Triples_Anotados               554
Triples_Intentados             554
Porcentaje_Triples            3238
Dobles_Anotados                554
Dobles_Intentados              554
Porcentaje_Dobles              738
Porcentaje_Efectivo_Tiro_Campo 652
```

8. Ejecución o transformación de datos

```python
num_cols = df_clean.select_dtypes(include=[np.number]).columns
cat_cols = df_clean.select_dtypes(exclude=[np.number]).columns

# Relleno numérico con medianas
medians = df_clean[num_cols].median(numeric_only=True) if len(num_cols) > 0 else pd.Series(dtype=float)
df_clean[num_cols] = df_clean[num_cols].fillna(medians)

# Relleno categórico con modas
if len(cat_cols) > 0:
    modes = df_clean[cat_cols].mode(dropna=True)
    if not modes.empty:
        mode_vals = modes.iloc[0]
        df_clean[cat_cols] = df_clean[cat_cols].fillna(mode_vals)

df_clean.isnull().sum() #Verificamos 0 en NAN

```

✓ 0.0s Python

9. Limpiar valores y eliminar texto

```python
#Limpiar valores numéricos
for col in num_cols:
    if df[col].isnull().sum() > 0:
        if df[col].skew() > 1:  # asimetría → usar mediana
            df[col] = df[col].fillna(df[col].median())
        else:  # simétrica → usar media
            df[col] = df[col].fillna(df[col].mean())
```

✓ 0.0s Python

```python
# Limpiar valores categóricos
for col in cat_cols:
    if df[col].isnull().sum() > 0:
        df[col] = df[col].fillna(df[col].mode()[0])# --- Limpiar valores categóricos ---
for col in cat_cols:
    if df[col].isnull().sum() > 0:
        df[col] = df[col].fillna(df[col].mode()[0])
```

✓ 0.1s Python

```python
#Eliminando texto inválido
for col in df.columns:
    invalids = df[df[col].astype(str).str.lower() == 'invalid_value'].shape[0]
    print(f"En la columna {col} hay {invalids} valores 'invalid_value'.")
```

✓ 0.5s Python

10. Ejecución de datos y limpiar valores

```python
# Ejecución o transformación de datos
num_cols = df.select_dtypes(include=['float64', 'int64', 'float32', 'int32']).columns
```

✓ 0.0s Python

```python
# Ejecución o transformación de datos
for col in num_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

✓ 0.0s Python

```python
# Limpiar valores nulos o faltantes
for col in num_cols:
    if df[col].isnull().sum() > 0:
        # Si toda la columna está vacía
        if df[col].dropna().empty:
            df[col].fillna(0, inplace=True)
        else:
            # Usar la mediana para valores faltantes
            df[col].fillna(df[col].median(), inplace=True)
```

✓ 0.0s Python

```python
# Ejecución o transformación de datos
cat_cols = df.select_dtypes(include=['object']).columns
```

✓ 0.0s Python

11. Limpiar valores y verificación

```python
# Limpiar valores nulos o faltantes
for col in cat_cols:
    if df[col].isnull().sum() > 0:
        modos = df[col].mode()  # obtener moda (valor más frecuente)
        if not modos.empty:
            # si existe una moda válida, usarla
            df[col].fillna(modos[0], inplace=True)
        else:
            # si toda la columna está vacía, asignar texto genérico
            df[col].fillna("Desconocido", inplace=True)
```

✓ 0.0s Python

Generate + Code + Markdown

```python
#Verificación final
df.isnull().sum()
```

✓ 0.0s Python

```
Rango                    0
Jugador                  0
Posición                 0
Edad                     0
Equipo                   0
Juegos                   0
Juegos_Iniciados         0
Minutos_Juego            0
Tiros_Campo_Anotados     0
Tiros_Campo_Intentados   0
Porcentaje_Tiros_Campo   0
```

12.Guardar el dataset limpio en un nuevo archivo CSV

```python
# Guardar el dataset limpio en un nuevo archivo CSV
df.to_csv("df_limpio.csv", index=False)
```

✓ 0.4s Python

Conclusiones

Al iniciar el análisis del CSV de la temporada 1997–1998 de la NBA, la base de datos presentaba varios problemas que dificultaban cualquier tipo de análisis serio. El primero era la presencia de valores nulos, especialmente en columnas relacionadas con el rendimiento y estadísticas de los jugadores, como minutos por partido, porcentaje de tiro, rebotes y asistencias. También se detectaron registros duplicados, lo que implicaba que algunos jugadores aparecían más de una vez. Otro problema relevante fue la existencia de tipos de datos incorrectos, como números almacenados como texto o porcentajes con caracteres especiales.

Para solucionarlo, apliqué distintas técnicas de limpieza y preparación de datos, evitando eliminar información valiosa. Primero exploré los datos usando df.info() e isnull().sum() para dimensionar los problemas. Luego realicé limpieza de texto, eliminando espacios sobrantes, corrigiendo cadenas vacías y normalizando los nombres de las columnas. Posteriormente convertí las columnas numéricas que estaban en formato texto a su tipo correcto y eliminé duplicados exactos usando drop_duplicates() para mantener registros únicos por jugador y temporada.

En lugar de borrar registros con datos faltantes, imputé valores nulos usando la mediana para variables numéricas y la moda para categóricas, conservando así la integridad de todas las columnas y la mayor parte de los registros. Al finalizar, validé los resultados con isnull().sum() y confirmé que no quedaban valores nulos ni duplicados. La base quedó completamente limpia, con tipos de datos correctos y lista para análisis confiables.

De este proyecto aprendí que la limpieza de datos es esencial en la ciencia de datos. Entendí que limpiar no significa eliminar, sino interpretar los datos, corregir errores y tomar decisiones estratégicas para mejorar su calidad. También comprendí que trabajar con datos reales requiere paciencia y atención al detalle, porque los problemas no siempre son evidentes. En conclusión, este proceso me enseñó a ser metódico y cuidadoso, a utilizar las herramientas de pandas de manera efectiva, y sobre todo, a valorar el poder que tiene una base de datos limpia para generar análisis confiables y significativos sobre el rendimiento de los jugadores en la NBA.