

Reporte Limpieza de Datos.



BUAP.



Santiago Adriaan Lazos Ovalle.
Introducción a la Ciencia de Datos.
Jaime Alejandro Romero Sierra.

Link al GitHub: <https://github.com/SantiagoLazos/NFL-Combine->

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 NFL Scouting Combine, un evento deportivo que se realiza cada año en Estados Unidos y que forma parte del proceso de reclutamiento de nuevos jugadores para la National Football League (NFL). Este evento reúne a los mejores prospectos universitarios de fútbol americano, quienes son evaluados por entrenadores, cazatalentos y directivos de los 32 equipos de la liga a través de una serie de pruebas físicas, técnicas y cognitivas.

El Combine representa uno de los momentos más importantes para los jugadores que buscan ingresar al nivel profesional, ya que los resultados de sus pruebas pueden influir directamente en su posición dentro del Draft de la NFL, donde los equipos eligen a los nuevos talentos.

Los datos que se generan en este evento son de enorme interés para analistas, entrenadores y especialistas en ciencia de datos deportiva, ya que permiten observar patrones de rendimiento, correlaciones entre medidas físicas y desempeño, y hasta predicciones sobre el éxito futuro de los atletas.

Para la creación de esta base, se utilizó información pública y estructurada con datos de distintos años del Combine. En algunos casos, los registros provienen de bases de datos deportivas abiertas, hojas de cálculo recopiladas por aficionados, y fuentes oficiales de la NFL combinadas con aportes de sitios como Pro Football Reference, y repositorios especializados en estadísticas atléticas.

El propósito fue reunir una base de datos realista, extensa y multidimensional, con la suficiente variedad de atributos como para aplicar un proceso de limpieza profesional.

Además, se eligió esta temática porque une dos áreas que me apasionan: el deporte y la ciencia de datos. Me pareció interesante analizar cómo la estadística y la programación pueden aplicarse en algo tan dinámico como el rendimiento físico de los jugadores, y cómo un dataset aparentemente “sucio” puede transformarse en una fuente útil para investigaciones deportivas, análisis de talento o evaluaciones de rendimiento atlético.

Descripción general del contenido

La base de datos contiene información detallada sobre los atletas que participaron en distintos años del NFL Combine. Cada registro corresponde a un jugador único en un año determinado, e incluye tanto sus datos personales como las mediciones obtenidas en las pruebas físicas realizadas durante el evento.

Las principales categorías de información son:

- Datos personales y de identificación: nombre del atleta, universidad de procedencia y posición de juego.
- Medidas corporales: altura, peso, tamaño de las manos y longitud del brazo, las cuales son importantes para evaluar la complexión física de los jugadores según su posición.
- Pruebas físicas y de desempeño: incluyen tiempos, saltos y repeticiones en ejercicios diseñados para medir velocidad, fuerza, explosividad y agilidad.
- Evaluaciones cognitivas: como el test Wonderlic, una prueba que mide la capacidad de razonamiento lógico y velocidad mental.

En términos de estructura, la base tiene 16 columnas y más de diez mil registros, lo cual permitió aplicar un proceso de limpieza significativo. Inicialmente se detectaron valores nulos, duplicados, nombres mal y formatos inconsistentes (por ejemplo, columnas numéricas almacenadas como texto).

El proceso de limpieza no se limitó a eliminar datos, sino que se centró en recuperar información y mejorar la coherencia. Se corrigieron tipos de datos, se imputaron valores faltantes con la mediana o la moda, y se eliminaron duplicados. El resultado fue una base limpia, uniforme y completamente funcional para su análisis posterior.

Este conjunto de datos no solo permite estudiar el desempeño físico de los atletas, sino también establecer comparaciones entre años, posiciones o universidades, abriendo la puerta a futuros proyectos de análisis predictivo o minería de datos aplicada al deporte.

Significado de cada columna

Columna	Descripción
year	Año en el que se realizó el Combine correspondiente al registro del jugador.
name	Nombre completo del atleta evaluado.
college	Universidad o institución donde jugó fútbol americano antes de participar en el Combine.
pos	Posición de juego del atleta (por ejemplo, QB = mariscal de campo, WR = receptor, RB = corredor, CB = esquinero, etc.).
height_(in)	Altura del jugador medida en pulgadas. Es una variable física clave para analizar ventajas posicionales.
weight_(lbs)	Peso corporal en libras. Refleja la masa y complejión del atleta.
hand_size_(in)	Tamaño de la mano del jugador en pulgadas, importante especialmente para posiciones como quarterback o receptor.
arm_length_(in)	Longitud del brazo medida en pulgadas; influye en el alcance y capacidad de bloqueo o recepción.
wonderlic	Puntuación obtenida en la prueba Wonderlic de razonamiento lógico y cognitivo.
40_yard	Tiempo (en segundos) en recorrer 40 yardas; mide la velocidad máxima del jugador.
bench_press	Número de repeticiones de press de banca con 225 libras; mide la fuerza superior del cuerpo.
vert_leap_(in)	Altura alcanzada en el salto vertical (en pulgadas); mide la potencia y explosividad.
broad_jump_(in)	Distancia alcanzada en el salto horizontal (en pulgadas); mide la fuerza y coordinación de piernas.
shuttle	Tiempo (en segundos) en la prueba de 20 yardas (agilidad lateral).
3cone	Tiempo (en segundos) en la prueba de 3 conos, que evalúa la rapidez y cambio de dirección.
60yd_shuttle	Tiempo (en segundos) en la prueba de 60 yardas, que mide resistencia y consistencia en agilidad prolongada.

Proceso de Limpieza

1. Cargar Librerías y Base de datos.

```
1 #Importamos las Librerías y Cargamos la Base de datos.
2 import pandas as pd
3 import numpy as np
4 df=pd.read_csv("C:/Users/salaz/OneDrive/Desktop/Ciencia de Datos/Proyecto NFL/Base de Datos Sucia.csv")
5 df
```

✓ 0.0s

	Year	Name	College	POS	Height (in)	Weight (lbs)	Hand Size (in)	Arm Length (in)	Wonderlic	40 Yard	Bench Press	Vert Leap (in)	Broad Jump (in)	Shuttle	3Cone	60Yd Shuttle
0	1987	Mike Adams	Arizona State	CB	NaN	198.0	8.50	30.50	NaN	4.42	13.0	32.0	118.0	4.60	NaN	Auto%#
1	1987	John Adickes	Auto%#	C	Auto%#	266.0	10.25	30.00	NaN	NaN	25.0	26.5	103.0	4.60	9.99	NaN
2	1987	Tommy Agee	Auburn	FB	NaN	217.0	9.00	30.75	NaN	9.99	15.0	NaN	NaN	9.99	9.99	NaN
3	1987	David Alexander	NaN	C	75.0	279.0	10.50	NaN	NaN	5.13	22.0	27.5	Auto%#	4.33	9.99	NaN
4	1987	Lyneal Alston	Southern Mississippi	WR	72.1	202.0	10.00	33.00	NaN	4.64	7.0	32.0	114.0	4.52	9.99	11.85
...
10687	2014	James Morris	Iowa	ILB	72.88	241.0	9.13	30.75	NaN	4.80	18.0	34.5	116.0	4.36	6.94	NaN
10688	2015	Tyler Kroft	Rutgers	TE	77.5	246.0	9.63	33.00	32.0	4.69	17.0	NaN	NaN	9.99	9.99	NaN
10689	1997	Daryl Terrell	Southern Mississippi	OG	76.5	307.0	10.88	33.63	NaN	5.23	28.0	29.0	103.0	4.77	8.33	NaN
10690	1991	Todd Scott	Southwestern Louisiana	CB	70.4	204.0	9.63	31.00	NaN	4.65	14.0	33.0	NaN	NaN	9.99	11.41
10691	1993	Mark Szlachcic	Bowling Green (OH)	WR	75.9	200.0	8.88	33.00	NaN	4.96	NaN	26.0	106.0	4.47	9.99	12.27

2. Buscamos Información Importante

Información General del DF, Valores y Tipos de Datos

```
1 #Con este comando encontramos la cantidad total de valores "NaN" por cada columna
2 df.isnull().sum()
```

✓ 0.0s

Outputs are collapsed ...

```
1 #Con este comando sacamos la informacion general
2 df.info()
```

✓ 0.0s

Outputs are collapsed ...

```
1 #Sacamos los nombres de las columnas seran utiles posteriormente
2 df.columns
```

✓ 0.0s

3. Creamos una Copia del Df.

```
1 df_clean = df.copy()
2 df_clean
```

✓ 0.0s

	Year	Name	College	POS	Height (in)	Weight (lbs)	Hand Size (in)	Arm Length (in)	Wonderlic	40 Yard	Bench Press	Vert Leap (in)	Broad Jump (in)	Shuttle	3Cone	60Yd Shuttle
0	1987	Mike Adams	Arizona State	CB	NaN	198.0	8.50	30.50	NaN	4.42	13.0	32.0	118.0	4.60	NaN	Auto%#
1	1987	John Adickes	Auto%#	C	Auto%#	266.0	10.25	30.00	NaN	NaN	25.0	26.5	103.0	4.60	9.99	NaN
2	1987	Tommy Agee	Auburn	FB	NaN	217.0	9.00	30.75	NaN	9.99	15.0	NaN	NaN	9.99	9.99	NaN
3	1987	David Alexander	NaN	C	75.0	279.0	10.50	NaN	NaN	5.13	22.0	27.5	Auto%#	4.33	9.99	NaN
4	1987	Lyneal Alston	Southern Mississippi	WR	72.1	202.0	10.00	33.00	NaN	4.64	7.0	32.0	114.0	4.52	9.99	11.85
...
10687	2014	James Morris	Iowa	ILB	72.88	241.0	9.13	30.75	NaN	4.80	18.0	34.5	116.0	4.36	6.94	NaN
10688	2015	Tyler Kroft	Rutgers	TE	77.5	246.0	9.63	33.00	32.0	4.69	17.0	NaN	NaN	9.99	9.99	NaN
10689	1997	Daryl Terrell	Southern Mississippi	OG	76.5	307.0	10.88	33.63	NaN	5.23	28.0	29.0	103.0	4.77	8.33	NaN
10690	1991	Todd Scott	Southwestern Louisiana	CB	70.4	204.0	9.63	31.00	NaN	4.65	14.0	33.0	NaN	NaN	9.99	11.41
10691	1993	Mark Szlachcic	Bowling Green (OH)	WR	75.9	200.0	8.88	33.00	NaN	4.96	NaN	26.0	106.0	4.47	9.99	12.27

10692 rows × 16 columns

4. Buscamos y quitamos los valores duplicados

```
Calcular Valores Duplicados

1 #Detectamos los valores duplicados
2 df.duplicated()
3 df.duplicated().sum()
59] ✓ 0.0s
.. np.int64(260)

5 Eliminar Duplicados
1 df_clean = df_clean.drop_duplicates()
2 df_clean.shape
60] ✓ 0.0s
.. (10432, 16)
```

5.Creamos un diccionario de Palabras a quitar del Df, y las cambiamos por nan

```
Diccionario de Palabras a quitar y remplazo por nan

1 palabras_a_eliminar = ["Auto%#"]
2 df_clean = df_clean.replace(palabras_a_eliminar, np.nan)
61] ✓ 0.0s
```

6. Limpieza de tipo Objeto

- Revisa si hay columnas de texto.
- Limpia los espacios de más dentro de esas columnas.
- Quita espacios al principio y al final.
- Convierte textos vacíos en nan

```
Limpiar Tipo Obj

1 obj_cols = df_clean.select_dtypes(include=["object"]).columns
2 obj_cols
:] ✓ 0.0s
Index(['Year', 'Name', 'College', 'POS', 'Height (in)', 'Vert Leap (in)',
       'Broad Jump (in)', '60Yd Shuttle'],
      dtype='object')

1 if len(obj_cols) > 0:
2     ...df_clean[obj_cols] = df_clean[obj_cols].apply(lambda s: s.str.replace(r"\s+", " ", regex=True).str.strip())
3     ...df_clean[obj_cols] = df_clean[obj_cols].replace({"": np.nan})
:] ✓ 0.0s
```

7. Limpieza de numeros

```
Limpiar Tipo Numericos

1 obj_cols = df_clean.select_dtypes(include=["object"]).columns
2
3 for col in obj_cols:
4     try:
5         df_clean[col] = pd.to_numeric(df_clean[col])
6     except:
7         pass
8
9 df_clean.dtypes
10
11 ✓ 0.0s
12
13 Year           float64
14 Name            object
15 College          object
16 POS              object
17 Height (in)      float64
18 Weight (lbs)      float64
19 Hand Size (in)    float64
20 Arm Length (in)   float64
21 Wonderlic       float64
22 40 Yard          float64
23 Bench Press       float64
24 Vert Leap (in)    float64
25 Broad Jump (in)   float64
26 Shuttle           float64
27 3Cone             float64
28 60Yd Shuttle       float64
29 dtype: object
30
```

8.Quitamos Nan

num_cols=mediana de la columna
cat_cols= moda de la columna

```
1 num_cols = df_clean.select_dtypes(include=[np.number]).columns
2 cat_cols = df_clean.select_dtypes(exclude=[np.number]).columns
3
4 # Relleno numérico con medianas
5 medians = df_clean[num_cols].median(numeric_only=True) if len(num_cols) > 0 else pd.Series(dtype=float)
6 df_clean[num_cols] = df_clean[num_cols].fillna(medians)
7
8 # Relleno categórico con modas
9 if len(cat_cols) > 0:
10     modes = df_clean[cat_cols].mode(dropna=True)
11     if not modes.empty:
12         mode_vals = modes.iloc[0]
13         df_clean[cat_cols] = df_clean[cat_cols].fillna(mode_vals)
14
15 df_clean.isnull().sum() #Verificamos 0 en NAN
16
17 ✓ 0.0s
18
```

9. Hacer una Copia y Guardar el csv limpio

```
Guardar el csv limpio

1 df_clean.to_csv("Base de Datos Limpia", index=False)
2
3 ✓ 0.1s
4
```

10. Comparar los csv visualmente y corroborar los cambios

CSV Sucio

```
1 #Importamos las Librerias y Cargamos la Base de datos.
2 import pandas as pd
3 import numpy as np
4 df=pd.read_csv("C:/Users/salaz/OneDrive/Desktop/Ciencia de Datos/Proyecto NFL/Base de Datos Sucia.csv")
5 df
```

✓ 0.0s

	Year	Name	College	POS	Height (in)	Weight (lbs)	Hand Size (in)	Arm Length (in)	Wonderlic	40 Yard	Bench Press	Vert Leap (in)	Broad Jump (in)	Shuttle	3Cone	60Yd Shuttle
0	1987	Mike Adams	Arizona State	CB	NaN	198.0	8.50	30.50	NaN	4.42	13.0	32.0	118.0	4.60	NaN	Auto%#
1	1987	John Adickes	Auto%#	C	Auto%#	266.0	10.25	30.00	NaN	NaN	25.0	26.5	103.0	4.60	9.99	NaN
2	1987	Tommy Agee	Auburn	FB	NaN	217.0	9.00	30.75	NaN	9.99	15.0	NaN	NaN	9.99	9.99	NaN
3	1987	David Alexander	NaN	C	75.0	279.0	10.50	NaN	NaN	5.13	22.0	27.5	Auto%#	4.33	9.99	NaN
4	1987	Lyneal Alston	Southern Mississippi	WR	72.1	202.0	10.00	33.00	NaN	4.64	7.0	32.0	114.0	4.52	9.99	11.85
...
10687	2014	James Morris	Iowa	ILB	72.88	241.0	9.13	30.75	NaN	4.80	18.0	34.5	116.0	4.36	6.94	NaN
10688	2015	Tyler Kroft	Rutgers	TE	77.5	246.0	9.63	33.00	32.0	4.69	17.0	NaN	NaN	9.99	9.99	NaN
10689	1997	Daryl Terrell	Southern Mississippi	OG	76.5	307.0	10.88	33.63	NaN	5.23	28.0	29.0	103.0	4.77	8.33	NaN
10690	1991	Todd Scott	Southwestern Louisiana	CB	70.4	204.0	9.63	31.00	NaN	4.65	14.0	33.0	NaN	NaN	9.99	11.41
10691	1993	Mark Szlachcic	Bowling Green (OH)	WR	75.9	200.0	8.88	33.00	NaN	4.96	NaN	26.0	106.0	4.47	9.99	12.27

10692 rows × 16 columns

CSV Limpio

```
1 import pandas as pd
2 df=pd.read_csv("C:/Users/salaz/OneDrive/Desktop/Ciencia de Datos/Proyecto NFL/Base de Datos Limpia")
3 df
```

✓ 0.0s

	Year	Name	College	POS	Height (in)	Weight (lbs)	Hand Size (in)	Arm Length (in)	Wonderlic	40 Yard	Bench Press	Vert Leap (in)	Broad Jump (in)	Shuttle	3Cone	60Yd Shuttle
0	1987.0	Mike Adams	Arizona State	CB	74.00	198.0	8.50	30.50	24.0	4.42	13.0	32.0	118.0	4.60	9.99	11.65
1	1987.0	John Adickes	Florida State	C	74.00	266.0	10.25	30.00	24.0	4.80	25.0	26.5	103.0	4.60	9.99	11.65
2	1987.0	Tommy Agee	Auburn	FB	74.00	217.0	9.00	30.75	24.0	9.99	15.0	32.0	113.0	9.99	9.99	11.65
3	1987.0	David Alexander	Florida State	C	75.00	279.0	10.50	32.25	24.0	5.13	22.0	27.5	113.0	4.33	9.99	11.65
4	1987.0	Lyneal Alston	Southern Mississippi	WR	72.10	202.0	10.00	33.00	24.0	4.64	7.0	32.0	114.0	4.52	9.99	11.85
...
10427	2015.0	Steve Smith	Michigan State	FS	72.63	208.0	10.38	32.25	24.0	4.65	20.0	32.0	119.0	4.33	7.09	11.65
10428	2001.0	Steve Smith	Penn State	CB	72.30	199.0	8.50	31.00	24.0	4.53	20.0	34.5	113.0	4.27	7.21	11.84
10429	2014.0	James Morris	Iowa	ILB	72.88	241.0	9.13	30.75	24.0	4.80	18.0	34.5	116.0	4.36	6.94	11.65
10430	1991.0	Todd Scott	Southwestern Louisiana	CB	70.40	204.0	9.63	31.00	24.0	4.65	14.0	33.0	113.0	4.52	9.99	11.41
10431	1993.0	Mark Szlachcic	Bowling Green (OH)	WR	75.90	200.0	8.88	33.00	24.0	4.96	20.0	26.0	106.0	4.47	9.99	12.27

10432 rows × 16 columns

Notamos que se reemplazaron los NaN y que desaparece los Auto%#

11. Chequeo isnan() de cvs Limpio

```
"C:/Users/salaz/OneDrive/Desktop/Ciencia de Datos/Proyecto NFL/Base de Datos Limpia"
```

```
1 df.isnull().sum()
✓ 0.0s
```

year	0
name	0
college	0
pos	0
height_(in)	0
weight_(lbs)	0
hand_size_(in)	0
arm_length_(in)	0
wonderlic	0
40_yard	0
bench_press	0
vert_leap_(in)	0
broad_jump_(in)	0
shuttle	0
3cone	0
60yd_shuttle	0
dtype:	int64

Conclusiones

Al iniciar el proyecto del NFL Combine, la base de datos presentaba varios problemas que dificultaban cualquier tipo de análisis serio. El primero era la gran cantidad de valores nulos, especialmente en las columnas relacionadas con las pruebas físicas, como *Wonderlic*, *Bench Press*, *Broad Jump* y *Hand Size*. También encontré registros duplicados, lo que significaba que algunos jugadores aparecían más de una vez. Otro problema importante fue la presencia de tipos de datos incorrectos, como números almacenados como texto o datos con caracteres especiales, y por último, la Palabra Auto%#.

Para solucionarlo, apliqué diferentes técnicas de limpieza y preparación de datos, siempre evitando el uso del comando prohibido `DROPNA()` como método principal, porque el objetivo era recuperar información y no eliminarla.

Primero, exploré los datos usando `DF.INFO` e `ISNULL.SUM()` para entender la magnitud de los problemas.

Después, realicé una limpieza de texto, eliminando espacios sobrantes, corrigiendo cadenas vacías y reemplazando palabras sin sentido por valores nulos. En la etapa siguiente, convertí las columnas numéricas que estaban en formato de texto a su tipo correcto. También eliminé duplicados exactos con `drop.duplicates` para mantener solo registros únicos por atleta y año.

Finalmente, en vez de borrar los registros con datos faltantes, imputé los valores nulos usando la mediana para las variables numéricas y la moda para las categóricas, logrando conservar el 100% de las columnas y la mayor parte de los registros.

Al terminar todo el proceso, validé los resultados con `ISNULL.SUM()` y comprobé que no quedaban valores nulos ni duplicados. La base quedó completamente limpia, con tipos de datos correctos y nombres normalizados.

De este proyecto aprendí que la limpieza de datos es una de las partes más importantes en la ciencia de datos. Entendí que limpiar no significa eliminar, sino entender los datos, reparar los errores y tomar decisiones informadas para mejorar su calidad. También me di cuenta de que trabajar con datos reales exige paciencia y atención al detalle, porque los errores no siempre son evidentes.

En conclusión, este proceso me enseñó a ser más cuidadoso y metódico, a utilizar las herramientas de `pandas` de manera estratégica, y sobre todo, a valorar el poder que tiene una base limpia para generar análisis confiables. Fue una experiencia que combinó lo técnico con lo analítico, y que me ayudó a entender

cómo la ciencia de datos puede aplicarse a algo que realmente me apasiona: el deporte y el rendimiento de los atletas.