

In [628...]

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import numpy as np
```

Loading the datasets

In [952...]

```
df_fights = pd.read_csv('./dataset/raw_data/ufc_events.csv')
df_fighters = pd.read_csv('./dataset/raw_data/ufc_fighters.csv')
```

Data Exploring and cleaning ufc fighters dataframe

In [953...]

```
df_fighters.head()
```

Out[953...]

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SAPM | StrDef |
|---|--|-----------------|---------|--------|----------|-------|----------|--------------|------|--------|-------|--------|
| 0 | | Tom Aaron | 5-3-0 | -- | 155 lbs. | -- | NaN | Jul 13, 1978 | 0.00 | 0% | 0.00 | 0% |
| 1 | | Danny Abbadi | 4-6-0 | 5' 11" | 155 lbs. | -- | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% |
| 2 | | Nariman Abbasov | 28-4-0 | 5' 8" | 155 lbs. | 66" | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% |
| 3 | | Darion Abbey | 9-5-0 | 6' 2" | 265 lbs. | 80" | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% |
| 4 | | David Abbott | 10-15-0 | 6' 0" | 265 lbs. | -- | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% |



In [954...]

```
df_fighters.shape
```

Out[954...]

```
(4449, 15)
```

In [955...]

```
df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4449 entries, 0 to 4448
Data columns (total 15 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   name      4449 non-null    object  
 1   record     4449 non-null    object  
 2   height     4449 non-null    object  
 3   weight     4449 non-null    object  
 4   reach      4449 non-null    object  
 5   stance     3601 non-null    object  
 6   dob        4449 non-null    object  
 7   SLpM       4449 non-null    float64 
 8   StrAcc     4449 non-null    object  
 9   SApM       4449 non-null    float64 
 10  StrDef     4449 non-null    object  
 11  TDAvg      4449 non-null    float64 
 12  TDAcc      4449 non-null    object  
 13  TDDef      4449 non-null    object  
 14  SubAvg     4449 non-null    float64 
dtypes: float64(4), object(11)
memory usage: 521.5+ KB
```

In [956... df_fighters.head()

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | StrDef |
|---|-----------------|-----------------|---------|--------|----------|-------|----------|--------------|------|--------|-------|--------|
| 0 | Tom Aaron | Tom Aaron | 5-3-0 | -- | 155 lbs. | -- | NaN | Jul 13, 1978 | 0.00 | 0% | 0.00 | 0% |
| 1 | Danny Abbadi | Danny Abbadi | 4-6-0 | 5' 11" | 155 lbs. | -- | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% |
| 2 | Nariman Abbasov | Nariman Abbasov | 28-4-0 | 5' 8" | 155 lbs. | 66" | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% |
| 3 | Darion Abbey | Darion Abbey | 9-5-0 | 6' 2" | 265 lbs. | 80" | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% |
| 4 | David Abbott | David Abbott | 10-15-0 | 6' 0" | 265 lbs. | -- | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% |



In [957... fighter_cols = df_fighters.columns.to_list()

In [958... fighter_cols

```
Out[958... ['name',
'record',
'height',
'weight',
'reach',
'stance',
'dob',
'SLpM',
'StrAcc',
'SApM',
'StrDef',
'TDAvg',
'TDAcc',
'TDDef',
'SubAvg']
```

check for rows which are missing all important features

```
In [959... df_fighters.replace("--", np.nan, inplace=True)
In [960... indexes_to_drop = df_fighters.loc[(df_fighters["reach"].isna())&(df_fighters["weigh
In [961... len(indexes_to_drop)
Out[961... 65
In [962... # Drop those rows
df_fighters.drop(indexes_to_drop, inplace=True)
In [963... df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4384 entries, 0 to 4448
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype  
---  -- 
 0   name    4384 non-null   object 
 1   record   4384 non-null   object 
 2   height   4129 non-null   object 
 3   weight   4363 non-null   object 
 4   reach    2507 non-null   object 
 5   stance   3601 non-null   object 
 6   dob     3935 non-null   object 
 7   SLpM    4384 non-null   float64
 8   StrAcc  4384 non-null   object 
 9   SApM    4384 non-null   float64
 10  StrDef  4384 non-null   object 
 11  TDAvg   4384 non-null   float64
 12  TDAcc   4384 non-null   object 
 13  TDDef   4384 non-null   object 
 14  SubAvg  4384 non-null   float64
dtypes: float64(4), object(11)
memory usage: 548.0+ KB
```

```
In [964...]: indexes_to_drop2 = df_fighters.loc[(df_fighters["SLpM"]==0)&(df_fighters["StrAcc"]==len(indexes_to_drop2))]
```

```
Out[964...]: 630
```

```
In [965...]: df_fighters.drop(indexes_to_drop2, inplace=True)
```

```
In [966...]: df_fighters.shape
```

```
Out[966...]: (3754, 15)
```

```
In [967...]: df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3754 entries, 1 to 4448
Data columns (total 15 columns):
 #   Column  Non-Null Count  Dtype  
---  -- 
 0   name     3754 non-null   object  
 1   record    3754 non-null   object  
 2   height    3681 non-null   object  
 3   weight    3742 non-null   object  
 4   reach     2488 non-null   object  
 5   stance    3408 non-null   object  
 6   dob       3558 non-null   object  
 7   SLpM      3754 non-null   float64 
 8   StrAcc    3754 non-null   object  
 9   SApM      3754 non-null   float64 
 10  StrDef    3754 non-null   object  
 11  TDAvg     3754 non-null   float64 
 12  TDAcc     3754 non-null   object  
 13  TDDef     3754 non-null   object  
 14  SubAvg    3754 non-null   float64 
dtypes: float64(4), object(11)
memory usage: 469.2+ KB
```

```
In [968...]: print("Null Value and zero counts for fighter metrics")
for i in fighter_cols:
    mask = (
        (df_fighters[i] == "0%") |
        (df_fighters[i] == 0) |
        (df_fighters[i].isna())
    )

    count = mask.sum()

    if count > 0:
        print(f"{i}: {count}")
```

```
Null Value and zero counts for fighter metrics
height: 73
weight: 12
reach: 1266
stance: 346
dob: 196
SLpM: 106
StrAcc: 106
SApM: 31
StrDef: 39
TDAvg: 1168
TDAcc: 1168
TDDef: 862
SubAvg: 1820
```

```
In [969...]: df_fighters = df_fighters.reset_index(drop=True)
```

```
In [970...]: df_fighters.head(20)
```

Out[970...]

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM |
|----|---------------------|------|------------------|--------|----------|-------|----------|--------------|------|--------|-------|
| 0 | Danny Abbadi | | 4-6-0 | 5' 11" | 155 lbs. | NaN | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 |
| 1 | Nariman Abbasov | | 28-4-0 | 5' 8" | 155 lbs. | 66" | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 |
| 2 | Darion Abbey | | 9-5-0 | 6' 2" | 265 lbs. | 80" | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 |
| 3 | David Abbott | | 10-15-0 | 6' 0" | 265 lbs. | NaN | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 |
| 4 | Hamdy Abdelwahab | | 7-1-0 (1 NC) | 6' 2" | 265 lbs. | 72" | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 |
| 5 | Mansur Abdul-Malik | | 8-0-1 | 6' 2" | 185 lbs. | 80" | Orthodox | Oct 07, 1997 | 4.27 | 48% | 3.49 |
| 6 | Shamil Abdurakhimov | | 20-8-0 | 6' 3" | 235 lbs. | 76" | Orthodox | Sep 02, 1981 | 2.41 | 44% | 3.02 |
| 7 | Hiroyuki Abe | | 8-15-3 (1 NC) | 5' 6" | 145 lbs. | NaN | Orthodox | Feb 09, 1970 | 1.71 | 36% | 3.11 |
| 8 | Daichi Abe | | 6-2-0 | 5' 11" | 170 lbs. | 71" | Orthodox | Nov 27, 1991 | 3.80 | 33% | 4.49 |
| 9 | Papy Abedi | | 10-4-0 | 5' 11" | 185 lbs. | NaN | Southpaw | Jun 30, 1978 | 2.80 | 55% | 3.15 |
| 10 | Ricardo Abreu | | 5-1-0 | 5' 11" | 185 lbs. | NaN | Orthodox | Apr 27, 1984 | 3.79 | 31% | 3.98 |
| 11 | Klidson Abreu | | 15-4-0 (1 NC) | 6' 0" | 205 lbs. | 74" | Orthodox | Dec 24, 1992 | 2.05 | 40% | 2.90 |
| 12 | Daniel Acacio | | 30-18-0 | 5' 8" | 180 lbs. | NaN | Orthodox | Dec 27, 1977 | 3.52 | 36% | 2.85 |
| 13 | John Adajar | | 6-2-0 | 5' 9" | 170 lbs. | 75" | Orthodox | Jun 22, 1991 | 3.90 | 52% | 6.28 |

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM |
|----|-----------------|------|--------|--------|-------------|-------|----------|--------------------|------|--------|------|
| 14 | Juan Adams | | 5-3-0 | 6' 5" | 265 lbs. | 80" | Orthodox | Jan 16, 1992 | 7.09 | 55% | 4.06 |
| 15 | Anthony Adams | | 8-2-0 | 6' 1" | 185 lbs. | 76" | Orthodox | Jan 13, 1988 | 3.17 | 41% | 5.93 |
| 16 | Zarrukh Adashev | | 4-4-0 | 5' 5" | 125 lbs. | 65" | Southpaw | Jul 29, 1992 | 3.65 | 40% | 3.04 |
| 17 | Israel Adesanya | | 24-5-0 | 6' 4" | 185 lbs. | 80" | Switch | Jul 22, 1989 | 4.02 | 48% | 3.20 |
| 18 | Mohamed Ado | | 5-1-0 | 5' 11" | 170 lbs. | 76" | Switch | May 03, 2000 | 1.66 | 83% | 0.33 |
| 19 | Nick Agallar | | 24-6-0 | 5' 8" | 155 lbs. | NaN | Orthodox | Jan 13, 1979 | 0.69 | 11% | 4.56 |

In [971]:

```
# check if there are fighters with the same name
df_fighters[df_fighters.duplicated(subset="name", keep=False)]
```

Out[971...]

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | St |
|------|--|-------------------|----------------|--------|----------|-------|----------|--------------|------|--------|------|----|
| 1166 | | Joey Gomez | 6-2-0 | 5' 10" | 135 lbs. | 73" | Orthodox | Jul 21, 1986 | 2.44 | 28% | 4.46 | |
| 1168 | | Joey Gomez | 7-1-0 | 5' 10" | 155 lbs. | 71" | Orthodox | Aug 29, 1989 | 3.73 | 49% | 3.33 | |
| 1551 | | Tony Johnson | 7-2-0 | 6' 2" | 205 lbs. | 76" | Orthodox | May 02, 1983 | 4.00 | 92% | 3.67 | |
| 1559 | | Tony Johnson | 11-3-0 | 6' 1" | 265 lbs. | NaN | NaN | NaN | 2.00 | 53% | 4.73 | |
| 2095 | | Michael McDonald | 1-1-0 | 5' 11" | 205 lbs. | NaN | Orthodox | Feb 06, 1965 | 0.00 | 0% | 0.40 | |
| 2096 | | Michael McDonald | 17-4-0 | 5' 9" | 135 lbs. | 70" | Orthodox | Jan 15, 1991 | 2.69 | 42% | 2.76 | |
| 3098 | | Jean Silva | 19-12-3 (1 NC) | 5' 6" | 160 lbs. | NaN | Orthodox | Oct 08, 1977 | 0.73 | 22% | 2.93 | |
| 3112 | | Bruno Silva | 15-7-2 (1 NC) | 5' 4" | 125 lbs. | 65" | Orthodox | Mar 16, 1990 | 3.82 | 50% | 4.55 | |
| 3113 | | Bruno Silva | 23-13-0 | 6' 0" | 185 lbs. | 74" | Orthodox | Jul 13, 1989 | 3.86 | 48% | 5.35 | |
| 3123 | | Jean Silva | 16-3-0 | 5' 7" | 145 lbs. | 69" | Orthodox | Dec 27, 1996 | 4.79 | 51% | 4.69 | |
| 3464 | | Victor Valenzuela | 13-6-2 | 5' 10" | 155 lbs. | NaN | NaN | NaN | 1.28 | 100% | 2.55 | |
| 3466 | | Victor Valenzuela | 12-4-0 | 5' 9" | 170 lbs. | 71" | Orthodox | Feb 09, 1994 | 3.36 | 33% | 8.40 | |



In [972...]

```
df_fighters.iloc[1168,0] = 'Joey Gomez 155'
df_fighters.iloc[1559,0] = 'Tony Johnson 265'
df_fighters.iloc[2096,0] = 'Michael McDonald 135'
df_fighters.iloc[3123,0] = 'Jean Silva 145'
df_fighters.iloc[3113,0] = 'Bruno Silva 185'
df_fighters.iloc[3466,0] = 'Victor Valenzuela 170'
```

In [973...]

```
df_fighters[df_fighters['name']=='Joey Gomez 155']
```

Out[973...]

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | StrDe |
|------|--|------------|--------|--------|----------|-------|----------|--------------|------|--------|------|-------|
| 1168 | | Joey Gomez | 7-1-0 | 5' 10" | 155 lbs. | 71" | Orthodox | Aug 29, 1989 | 3.73 | 49% | 3.33 | 50% |



In [974...]

```
# check if there are fighters with the same name
df_fighters[df_fighters.duplicated(subset="name", keep=False)]
```

Out[974...]

| | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg |
|--|------|--------|--------|--------|-------|--------|-----|------|--------|------|--------|-------|
| | | | | | | | | | | | | |



In [975...]

```
df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3754 entries, 0 to 3753
Data columns (total 15 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   name     3754 non-null   object  
 1   record    3754 non-null   object  
 2   height    3681 non-null   object  
 3   weight    3742 non-null   object  
 4   reach     2488 non-null   object  
 5   stance    3408 non-null   object  
 6   dob       3558 non-null   object  
 7   SLpM      3754 non-null   float64 
 8   StrAcc    3754 non-null   object  
 9   SApM      3754 non-null   float64 
 10  StrDef    3754 non-null   object  
 11  TDAvg     3754 non-null   float64 
 12  TDAcc     3754 non-null   object  
 13  TDDef     3754 non-null   object  
 14  SubAvg    3754 non-null   float64 
dtypes: float64(4), object(11)
memory usage: 440.1+ KB
```

In [976...]

```
def printNull(cls):

    print("Null Value and zero counts for fighter metrics")
    for i in cls:
        mask = (
            df_fighters[i].isna())

        )

        count = mask.sum()

        if count > 0:
            print(f"{i}: {count}")
```

In [977...]

```
printNull(fighter_cols)
```

Null Value and zero counts for fighter metrics

height: 73

weight: 12

reach: 1266

stance: 346

dob: 196

In [978...]

```
df_fighters[df_fighters['height'].isna()]
```

Out[978...]

| | | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM |
|------|-----|--------------------|--------------|--------|----------|-------|----------|--------------|------|--------|------|
| 42 | | Israel Albuquerque | 0-3-0 | NaN | 185 lbs. | NaN | Orthodox | NaN | 0.32 | 13% | 3.34 |
| 129 | | Angelo Antonio | 1-4-0 | NaN | 205 lbs. | NaN | Orthodox | Oct 04, 1998 | 1.09 | 50% | 7.64 |
| 316 | | Tom Blackledge | 10-7-0 | NaN | 205 lbs. | NaN | Orthodox | NaN | 2.91 | 61% | 2.55 |
| 382 | | Mack Brewer | 0-1-0 | NaN | 205 lbs. | NaN | NaN | NaN | 0.00 | 0% | 6.67 |
| 492 | | Remo Cardarelli | 7-5-0 | NaN | 135 lbs. | NaN | NaN | NaN | 2.27 | 47% | 1.47 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 3584 | | Sam Watford | 2-2-0 | NaN | 155 lbs. | NaN | NaN | NaN | 0.13 | 18% | 0.73 |
| 3585 | | Blake Watkins | 5-1-0 | NaN | 185 lbs. | NaN | NaN | NaN | 5.45 | 62% | 7.64 |
| 3587 | | Maka Watson | 4-2-0 | NaN | 155 lbs. | NaN | NaN | Sep 22, 1984 | 0.93 | 37% | 1.60 |
| 3605 | | Aaron Wetherspoon | 9-5-0 (1 NC) | NaN | 170 lbs. | NaN | NaN | Jan 11, 1979 | 2.27 | 32% | 4.33 |
| 3746 | | Mike Zichelle | 8-4-0 | NaN | 205 lbs. | NaN | NaN | NaN | 0.87 | 35% | 0.87 |

73 rows × 15 columns



In [979...]

```
df_fighters.head()
```

Out[979...]

| | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | Strl |
|----------|------------------|-----------------|---------------|---------------|--------------|---------------|--------------|-------------|---------------|-------------|-------------|
| 0 | Danny Abbadi | 4-6-0 | 5' 11" | 155 lbs. | NaN | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 5 |
| 1 | Nariman Abbasov | 28-4-0 | 5' 8" | 155 lbs. | 66" | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 4 |
| 2 | Darion Abbey | 9-5-0 | 6' 2" | 265 lbs. | 80" | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 2 |
| 3 | David Abbott | 10-15-0 | 6' 0" | 265 lbs. | NaN | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 3 |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 6' 2" | 265 lbs. | 72" | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | 5 |



In [980...]

```
#Reach to centimeters
df_fighters['reach'] = df_fighters['reach'].astype(str).str.replace('\'', '')
df_fighters['reach'] = pd.to_numeric(df_fighters['reach'], errors='coerce')
df_fighters['reach'] = df_fighters['reach']*2.54

# fill reach ~ height * 1.06
#df['height_in'] = df['height'].str.extract(r"(\d+").astype(float) * 12 + df['height'].str[-2]
#df['reach'] = df['reach'].fillna(df['height_in'] * 1.06)
```

In [981...]

```
df_fighters.head()
```

Out[981...]

| | name | record | height | weight | reach | stance | dob | SLpM | StrAcc | SApM | Str |
|---|------------------|-----------------|--------|----------|--------|----------|--------------|------|--------|-------|-----|
| 0 | Danny Abbadi | 4-6-0 | 5' 11" | 155 lbs. | NaN | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | . |
| 1 | Nariman Abbasov | 28-4-0 | 5' 8" | 155 lbs. | 167.64 | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | . |
| 2 | Darion Abbey | 9-5-0 | 6' 2" | 265 lbs. | 203.20 | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | . |
| 3 | David Abbott | 10-15-0 | 6' 0" | 265 lbs. | NaN | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | . |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 6' 2" | 265 lbs. | 182.88 | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | . |



In [982...]

```
def height_to_cm(h):
    if pd.isna(h):
        return np.nan
    match = re.match(r"(\d+)' (\d+)", h)
    if match:
        feet = int(match.group(1))
        inches = int(match.group(2))
        total_inches = feet * 12 + inches
        return total_inches * 2.54
    return np.nan

df_fighters['height_cm'] = df_fighters['height'].apply(height_to_cm)
```

In [983...]

```
df_fighters.drop(columns="height", inplace=True)
```

In [984...]

```
df_fighters.head()
```

Out[984...]

| | name | record | weight | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TD |
|----------|------------------|---------------|---------------|--------------|---------------|--------------|-------------|---------------|-------------|---------------|-----------|
| 0 | Danny Abbadi | 4-6-0 | 155 lbs. | NaN | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% | |
| 1 | Nariman Abbasov | 28-4-0 | 155 lbs. | 167.64 | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% | |
| 2 | Darion Abbey | 9-5-0 | 265 lbs. | 203.20 | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% | |
| 3 | David Abbott | 10-15-0 | 265 lbs. | NaN | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% | |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 265 lbs. | 182.88 | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | 51% | |



In [985...]

```
#common ufc fighter height reach ratio
df_fighters['reach'] = df_fighters['reach'].fillna(df_fighters['height_cm'] * 1.06)
```

In [986...]

```
cols_updated = df_fighters.columns.to_list()
printNull(cols_updated)
```

Null Value and zero counts for fighter metrics
 weight: 12
 reach: 72
 stance: 346
 dob: 196
 height_cm: 73

In [987...]

```
df_fighters.head()
```

Out[987...]

| | name | record | weight | reach | stance | dob | SLpM | StrAcc | SApM | StrDef |
|---|------------------|-----------------|----------|----------|----------|--------------|------|--------|-------|--------|
| 0 | Danny Abbadi | 4-6-0 | 155 lbs. | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% |
| 1 | Nariman Abbasov | 28-4-0 | 155 lbs. | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% |
| 2 | Darion Abbey | 9-5-0 | 265 lbs. | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% |
| 3 | David Abbott | 10-15-0 | 265 lbs. | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 265 lbs. | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | 51% |



In [988...]

```
df_fighters['stance'] = df_fighters['stance'].fillna("unknown")
```

In [989...]

```
printNull(cols_updated)
```

Null Value and zero counts for fighter metrics
 weight: 12
 reach: 72
 dob: 196
 height_cm: 73

In [990...]

```
df_fighters.drop(df_fighters[df_fighters["weight"].isna()].index, inplace=True)
```

In [991...]

```
printNull(cols_updated)
```

Null Value and zero counts for fighter metrics
 reach: 61
 dob: 190
 height_cm: 62

In [992...]

```
df_fighters['height_cm'] = df_fighters['height_cm'].fillna(df_fighters['reach'] / 1)
```

In [993...]

```
printNull(cols_updated)
```

Null Value and zero counts for fighter metrics
 reach: 61
 dob: 190
 height_cm: 61

In [994...]

```
df_fighters["weight"] = (
    df_fighters["weight"]
    .str.replace("lbs.", "", regex=False)
    .str.replace("lbs", "", regex=False)
    .str.strip()
```

```
        .astype(float)
)
```

```
In [995... df_fighters["weight_kg"] = df_fighters["weight"] * 0.453592
```

```
In [996... df_fighters.drop(columns="weight", inplace=True)
```

```
In [997... df_fighters.head()
```

| | | name | record | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | 1 |
|---|------------------|-----------------|----------|----------|--------------|------|------|--------|------|--------|-------|---|
| 0 | Danny Abbadi | 4-6-0 | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% | 0.00 | | |
| 1 | Nariman Abbasov | 28-4-0 | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% | 0.00 | | |
| 2 | Darion Abbey | 9-5-0 | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% | 0.00 | | |
| 3 | David Abbott | 10-15-0 | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% | 1.07 | | |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | 51% | 3.25 | | |



```
In [998... cols_updated1 = df_fighters.columns.to_list()
printNull(cols_updated1)
```

Null Value and zero counts for fighter metrics
 reach: 61
 dob: 190
 height_cm: 61

```
In [999... df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3742 entries, 0 to 3753
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3742 non-null    object  
 1   record       3742 non-null    object  
 2   reach        3681 non-null    float64 
 3   stance       3742 non-null    object  
 4   dob          3552 non-null    object  
 5   SLpM         3742 non-null    float64 
 6   StrAcc       3742 non-null    object  
 7   SApM         3742 non-null    float64 
 8   StrDef       3742 non-null    object  
 9   TDAvg        3742 non-null    float64 
 10  TDAcc        3742 non-null    object  
 11  TDDef        3742 non-null    object  
 12  SubAvg       3742 non-null    float64 
 13  height_cm    3681 non-null    float64 
 14  weight_kg    3742 non-null    float64 
dtypes: float64(7), object(8)
memory usage: 467.8+ KB
```

```
In [100...]: df_fighters.drop(df_fighters[df_fighters['reach'].isna()].index,inplace=True)
```

```
In [100...]: printNull(cols_updated1)
```

```
Null Value and zero counts for fighter metrics
dob: 150
```

```
In [100...]: df_fighters.drop(df_fighters[df_fighters['dob'].isna()].index,inplace=True)
```

```
In [100...]: printNull(cols_updated1)
```

```
Null Value and zero counts for fighter metrics
```

```
In [100...]: df_fighters.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3531 entries, 0 to 3753
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3531 non-null   object  
 1   record       3531 non-null   object  
 2   reach        3531 non-null   float64 
 3   stance       3531 non-null   object  
 4   dob          3531 non-null   object  
 5   SLpM         3531 non-null   float64 
 6   StrAcc       3531 non-null   object  
 7   SApM         3531 non-null   float64 
 8   StrDef       3531 non-null   object  
 9   TDAvg        3531 non-null   float64 
 10  TDAcc        3531 non-null   object  
 11  TDDef        3531 non-null   object  
 12  SubAvg       3531 non-null   float64 
 13  height_cm    3531 non-null   float64 
 14  weight_kg    3531 non-null   float64 
dtypes: float64(7), object(8)
memory usage: 441.4+ KB
```

In [100...]: df_fighters.head()

Out[100...]:

| | | name | record | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | 1 |
|---|------------------|-----------------|--------|----------|----------|--------------|------|--------|-------|--------|-------|---|
| 0 | Danny Abbadi | | 4-6-0 | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 38% | 4.41 | 57% | 0.00 | |
| 1 | Nariman Abbasov | | 28-4-0 | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 20% | 5.67 | 46% | 0.00 | |
| 2 | Darion Abbey | | 9-5-0 | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 50% | 14.06 | 28% | 0.00 | |
| 3 | David Abbott | 10-15-0 | | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 30% | 3.55 | 38% | 1.07 | |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 55% | 3.67 | 51% | 3.25 | |



In [100...]:

```
percent_cols = ["StrAcc", "StrDef", "TDAcc", "TDDef"]

for col in percent_cols:
    df_fighters[col] = (
        df_fighters[col]
        .str.replace("%", "", regex=False)
        .str.replace("--", "", regex=False))
```

```
)  
df_fighters[col] = pd.to_numeric(df_fighters[col], errors="coerce") / 100
```

In [100...]: df_fighters.head()

Out[100...]:

| | | name | record | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc |
|---|------------------|-----------------|----------|----------|--------|--------------|------|--------|-------|--------|-------|-------|
| | | | | | | | | | | | | |
| 0 | Danny Abbadi | 4-6-0 | 191.1604 | Orthodox | | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 |
| 1 | Nariman Abbasov | 28-4-0 | 167.6400 | Orthodox | | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 |
| 2 | Darion Abbey | 9-5-0 | 203.2000 | Orthodox | | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 |
| 3 | David Abbott | 10-15-0 | 193.8528 | Switch | | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.00 |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 182.8800 | Southpaw | | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.00 |



In [100...]: df_fighters.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 3531 entries, 0 to 3753
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3531 non-null    object 
 1   record       3531 non-null    object 
 2   reach        3531 non-null    float64
 3   stance       3531 non-null    object 
 4   dob          3531 non-null    object 
 5   SLpM         3531 non-null    float64
 6   StrAcc       3531 non-null    float64
 7   SApM         3531 non-null    float64
 8   StrDef       3531 non-null    float64
 9   TDAvg        3531 non-null    float64
 10  TDAcc        3531 non-null    float64
 11  TDDef        3531 non-null    float64
 12  SubAvg       3531 non-null    float64
 13  height_cm    3531 non-null    float64
 14  weight_kg    3531 non-null    float64
dtypes: float64(11), object(4)
memory usage: 441.4+ KB
```

In [100...]: # split wins losses and draws, NC s seperately

```
def split_record(record):
    if pd.isna(record):
```

```

        return pd.Series([None, None, None, None])

    # Extract: wins-losses-draws
    base = record.split("(")[0].strip()           # "7-1-0" or "4-6-0"
    wins, losses, draws = base.split("-")

    # Extract NC if present
    nc_match = re.search(r"\((\d+)\s*NC\)", record)
    nc = int(nc_match.group(1)) if nc_match else 0

    return pd.Series([int(wins), int(losses), int(draws), nc])

# ----- APPLY TO DATAFRAME -----
df_fighters[["wins", "losses", "draws", "nc"]] = df_fighters["record"].apply(split_

```

In [101...]: df_fighters.head()

Out[101...]:

| | name | record | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | 1 |
|---|------------------|-----------------|----------|----------|--------------|------|--------|-------|--------|-------|---|
| 0 | Danny Abbadi | 4-6-0 | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | |
| 1 | Nariman Abbasov | 28-4-0 | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | |
| 2 | Darion Abbey | 9-5-0 | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | |
| 3 | David Abbott | 10-15-0 | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | |
| 4 | Hamdy Abdelwahab | 7-1-0 (1 NC) | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | |



In [101...]: df_fighters.drop(columns='record', inplace=True)

In [101...]: df_fighters.head()

Out[101...]

| | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc | 1 |
|----------|------------------|--------------|---------------|--------------|-------------|---------------|-------------|---------------|--------------|--------------|----------|
| 0 | Danny Abbadi | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 | |
| 1 | Nariman Abbasov | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 | |
| 2 | Darion Abbey | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 | |
| 3 | David Abbott | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.33 | |
| 4 | Hamdy Abdelwahab | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.65 | |



In [101...]

df_fighters.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 3531 entries, 0 to 3753
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3531 non-null   object 
 1   reach       3531 non-null   float64
 2   stance      3531 non-null   object 
 3   dob         3531 non-null   object 
 4   SLpM        3531 non-null   float64
 5   StrAcc      3531 non-null   float64
 6   SApM        3531 non-null   float64
 7   StrDef      3531 non-null   float64
 8   TDAvg       3531 non-null   float64
 9   TDAcc       3531 non-null   float64
 10  TDDef       3531 non-null   float64
 11  SubAvg      3531 non-null   float64
 12  height_cm   3531 non-null   float64
 13  weight_kg   3531 non-null   float64
 14  wins         3531 non-null   int64  
 15  losses        3531 non-null   int64  
 16  draws         3531 non-null   int64  
 17  nc            3531 non-null   int64  
dtypes: float64(11), int64(4), object(3)
memory usage: 524.1+ KB
```

In [101...]

```
def dob_to_age(dob):
    if pd.isna(dob):
        return None
    try:
        birthdate = pd.to_datetime(dob)
```

```

        today = pd.Timestamp.today()
        age = today.year - birthdate.year - ((today.month, today.day) < (birthdate.month, birthdate.day))
        return age
    except:
        return None

df_fighters["age"] = df_fighters["dob"].apply(dob_to_age)

```

In [101... df_fighters.head()

| | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc | 1 |
|---|------------------|----------|----------|--------------|------|--------|-------|--------|-------|-------|---|
| 0 | Danny Abbadi | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 | |
| 1 | Nariman Abbasov | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 | |
| 2 | Darion Abbey | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 | |
| 3 | David Abbott | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.33 | |
| 4 | Hamdy Abdelwahab | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.65 | |



In [101... df_fighters.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 3531 entries, 0 to 3753
Data columns (total 19 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3531 non-null    object  
 1   reach       3531 non-null    float64 
 2   stance      3531 non-null    object  
 3   dob         3531 non-null    object  
 4   SLpM        3531 non-null    float64 
 5   StrAcc      3531 non-null    float64 
 6   SApM        3531 non-null    float64 
 7   StrDef      3531 non-null    float64 
 8   TDAvg       3531 non-null    float64 
 9   TDAcc       3531 non-null    float64 
 10  TDDef       3531 non-null    float64 
 11  SubAvg      3531 non-null    float64 
 12  height_cm   3531 non-null    float64 
 13  weight_kg   3531 non-null    float64 
 14  wins         3531 non-null    int64   
 15  losses       3531 non-null    int64   
 16  draws         3531 non-null    int64   
 17  nc            3531 non-null    int64   
 18  age           3531 non-null    int64   
dtypes: float64(11), int64(5), object(3)
memory usage: 551.7+ KB
```

```
In [101...]: df_fighters["total_fights"] = df_fighters["wins"] + df_fighters["losses"] + df_fighters["draws"]
df_fighters["win_rate"] = df_fighters["wins"] / df_fighters["total_fights"]
```

```
In [101...]: df_fighters.head()
```

Out[101...]

| | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc | . |
|----------|------------------|--------------|---------------|--------------|-------------|---------------|-------------|---------------|--------------|--------------|---|
| 0 | Danny Abbadi | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 | . |
| 1 | Nariman Abbasov | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 | . |
| 2 | Darion Abbey | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 | . |
| 3 | David Abbott | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.33 | . |
| 4 | Hamdy Abdelwahab | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.65 | . |

5 rows × 21 columns



In [101...]

```
df_fighters.loc[(df_fighters['win_rate']==1) & (df_fighters['total_fights']>10)]
```

Out[101...]

| | | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TI |
|-------------|--------------------|----------|----------|--------|--------------|-------|--------|------|--------|-------|----|
| 62 | Nurullo Aliev | 182.8800 | Orthodox | | Dec 08, 2000 | 2.72 | 0.44 | 1.63 | 0.58 | 2.12 | |
| 244 | Farid Basharat | 180.3400 | Orthodox | | Aug 02, 1997 | 4.05 | 0.52 | 2.47 | 0.60 | 3.60 | |
| 560 | Khamzat Chimaev | 190.5000 | Orthodox | | May 01, 1994 | 4.04 | 0.60 | 2.32 | 0.43 | 5.29 | |
| 936 | Rafael Estevam | 175.2600 | Orthodox | | Aug 10, 1996 | 2.10 | 0.54 | 2.59 | 0.47 | 6.01 | |
| 948 | Movsar Evloev | 182.8800 | Orthodox | | Feb 11, 1994 | 3.99 | 0.48 | 2.66 | 0.60 | 4.67 | |
| 1003 | Torrez Finney | 190.5000 | Orthodox | | Oct 24, 1998 | 1.77 | 0.68 | 1.74 | 0.48 | 9.31 | |
| 1102 | Tommy Gantt | 193.0400 | Orthodox | | Jan 09, 1993 | 1.08 | 1.00 | 0.36 | 0.66 | 10.78 | |
| 1539 | Ramiro Jimenez | 185.4200 | Southpaw | | Jan 16, 1999 | 14.22 | 0.65 | 2.43 | 0.82 | 5.20 | |
| 1668 | Aliaskhab Khizriev | 187.9600 | Southpaw | | Aug 24, 1990 | 6.01 | 0.64 | 4.38 | 0.61 | 5.64 | |
| 1877 | Andre Lima | 170.1800 | Orthodox | | Feb 04, 1999 | 4.06 | 0.60 | 2.49 | 0.57 | 1.36 | |
| 2169 | Phillip Miller | 191.1604 | Orthodox | | Feb 20, 1979 | 1.61 | 0.54 | 1.93 | 0.50 | 3.62 | |
| 2220 | Muhammad Mokaev | 177.8000 | Orthodox | | Jul 30, 2000 | 1.61 | 0.45 | 1.35 | 0.57 | 5.29 | |
| 2251 | Michael Morales | 200.6600 | Orthodox | | Jun 24, 1999 | 5.68 | 0.49 | 3.26 | 0.53 | 1.04 | |
| 2299 | Samandar Murodov | 190.5000 | Orthodox | | Aug 28, 1999 | 1.50 | 0.75 | 0.50 | 0.44 | 6.01 | |

| | | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TI |
|-------------|--|---------------------|--------------|---------------|--------------|-------------|---------------|-------------|---------------|--------------|-----------|
| 2309 | | Azamat Murzakanov | 180.3400 | Southpaw | Apr 12, 1989 | 4.70 | 0.57 | 2.86 | 0.60 | 0.55 | |
| 2416 | | Khabib Nurmagomedov | 177.8000 | Orthodox | Sep 20, 1988 | 4.10 | 0.48 | 1.75 | 0.65 | 5.32 | |
| 2638 | | Mario Pinto | 200.6600 | Orthodox | Mar 14, 1998 | 3.39 | 0.78 | 2.42 | 0.53 | 2.72 | |
| 2709 | | Shavkat Rakhmonov | 195.5800 | Orthodox | Oct 23, 1994 | 3.25 | 0.60 | 2.17 | 0.51 | 1.40 | |
| 2779 | | Luke Riley | 175.2600 | Orthodox | Jun 30, 1999 | 2.55 | 0.60 | 1.27 | 0.46 | 0.00 | |
| 3164 | | Jacobe Smith | 182.8800 | Southpaw | Jan 17, 1996 | 6.21 | 0.66 | 2.86 | 0.47 | 7.42 | |
| 3191 | | Dmitry Sosnovskiy | 195.5800 | Orthodox | Jul 06, 1989 | 2.95 | 0.70 | 2.53 | 0.35 | 3.16 | |
| 3289 | | Baisangur Susurkaev | 200.6600 | Orthodox | Jan 08, 2001 | 4.56 | 0.59 | 3.78 | 0.53 | 2.76 | |
| 3397 | | Ilia Topuria | 175.2600 | Orthodox | Jan 21, 1997 | 4.81 | 0.48 | 3.83 | 0.64 | 1.96 | |

23 rows × 21 columns

In [102...]

df_fighters.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 3531 entries, 0 to 3753
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   name        3531 non-null    object  
 1   reach       3531 non-null    float64 
 2   stance      3531 non-null    object  
 3   dob         3531 non-null    object  
 4   SLpM        3531 non-null    float64 
 5   StrAcc      3531 non-null    float64 
 6   SApM        3531 non-null    float64 
 7   StrDef      3531 non-null    float64 
 8   TDAvg       3531 non-null    float64 
 9   TDAcc       3531 non-null    float64 
 10  TDDef       3531 non-null    float64 
 11  SubAvg      3531 non-null    float64 
 12  height_cm   3531 non-null    float64 
 13  weight_kg   3531 non-null    float64 
 14  wins         3531 non-null    int64   
 15  losses        3531 non-null    int64   
 16  draws         3531 non-null    int64   
 17  nc            3531 non-null    int64   
 18  age           3531 non-null    int64   
 19  total_fights 3531 non-null    int64   
 20  win_rate      3531 non-null    float64 
dtypes: float64(12), int64(6), object(3)
memory usage: 606.9+ KB
```

```
In [102...]: df_fighters["stance_encoded"] = df_fighters["stance"].astype("category").cat.codes
```

```
In [102...]: df_fighters.head()
```

Out[102...]

| | name | reach | stance | dob | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc | . |
|----------|------------------|--------------|---------------|--------------|-------------|---------------|-------------|---------------|--------------|--------------|---|
| 0 | Danny Abbadi | 191.1604 | Orthodox | Jul 03, 1983 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 | . |
| 1 | Nariman Abbasov | 167.6400 | Orthodox | Feb 01, 1994 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 | . |
| 2 | Darion Abbey | 203.2000 | Orthodox | Feb 25, 1993 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 | . |
| 3 | David Abbott | 193.8528 | Switch | Apr 26, 1965 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.33 | . |
| 4 | Hamdy Abdelwahab | 182.8800 | Southpaw | Jan 22, 1993 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.65 | . |

5 rows × 22 columns



In [102...]

df_fighters1 = df_fighters.drop(columns=['dob', 'stance'])

In [102...]

df_fighters1.head()

Out[102...]

| | name | reach | SLpM | StrAcc | SApM | StrDef | TDAvg | TDAcc | TDDef | SubAvg | . |
|----------|------------------|--------------|-------------|---------------|-------------|---------------|--------------|--------------|--------------|---------------|---|
| 0 | Danny Abbadi | 191.1604 | 3.29 | 0.38 | 4.41 | 0.57 | 0.00 | 0.00 | 0.77 | 0.0 | . |
| 1 | Nariman Abbasov | 167.6400 | 3.00 | 0.20 | 5.67 | 0.46 | 0.00 | 0.00 | 0.66 | 0.0 | . |
| 2 | Darion Abbey | 203.2000 | 8.44 | 0.50 | 14.06 | 0.28 | 0.00 | 0.00 | 0.00 | 0.0 | . |
| 3 | David Abbott | 193.8528 | 1.35 | 0.30 | 3.55 | 0.38 | 1.07 | 0.33 | 0.66 | 0.0 | . |
| 4 | Hamdy Abdelwahab | 182.8800 | 4.27 | 0.55 | 3.67 | 0.51 | 3.25 | 0.65 | 1.00 | 0.0 | . |



UFC Fights dataframe pre processing

In [102...]

df_fights.head()

Out[102...]

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighte |
|----------|---------------------|------------------|-------------------|-----------------|-----------------|---------------------|---------------------|---------------|
| 0 | Arman Tsarukyan | Dan Hooker | November 22, 2025 | Doha, Qatar | Marc Goddard | 0 | 0 | |
| 1 | Ian Machado Garry | Belal Muhammad | November 22, 2025 | Doha, Qatar | Rich Mitchell | 0 | 0 | |
| 2 | Volkan Oezdemir | Alonzo Menifield | November 22, 2025 | Doha, Qatar | Marc Goddard | 1 | 0 | |
| 3 | Myktybek Orolbai | Jack Hermansson | November 22, 2025 | Doha, Qatar | Daniel Movahedi | 1 | 0 | |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | November 22, 2025 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | |



In [102...]

df_fights.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8208 entries, 0 to 8207
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        8208 non-null   object 
 1   fighter_2        8208 non-null   object 
 2   date             8208 non-null   object 
 3   location          8208 non-null   object 
 4   referee           8183 non-null   object 
 5   fighter_1_kd      8208 non-null   int64  
 6   fighter_2_kd      8208 non-null   int64  
 7   fighter_1_sig_str 8208 non-null   int64  
 8   fighter_2_sig_str 8208 non-null   int64  
 9   fighter_1_td       8208 non-null   int64  
 10  fighter_2_td       8208 non-null   int64  
 11  fighter_1_sub      8208 non-null   int64  
 12  fighter_2_sub      8208 non-null   int64  
 13  result            8208 non-null   object 
 14  winner             8208 non-null   object 
 15  weight_class       8208 non-null   object 
 16  method              8208 non-null   object 
 17  rounds              8208 non-null   int64  
 18  time                8208 non-null   object 
 19  title_fight         8208 non-null   bool  
dtypes: bool(1), int64(9), object(10)
memory usage: 1.2+ MB
```

In [102...]

df_fights.drop(df_fights[df_fights['referee'].isna()].index, inplace=True)

In [102...]

df_fights.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 8183 entries, 0 to 8207
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        8183 non-null    object 
 1   fighter_2        8183 non-null    object 
 2   date              8183 non-null    object 
 3   location          8183 non-null    object 
 4   referee            8183 non-null    object 
 5   fighter_1_kd      8183 non-null    int64  
 6   fighter_2_kd      8183 non-null    int64  
 7   fighter_1_sig_str 8183 non-null    int64  
 8   fighter_2_sig_str 8183 non-null    int64  
 9   fighter_1_td      8183 non-null    int64  
 10  fighter_2_td      8183 non-null    int64  
 11  fighter_1_sub     8183 non-null    int64  
 12  fighter_2_sub     8183 non-null    int64  
 13  result             8183 non-null    object 
 14  winner             8183 non-null    object 
 15  weight_class       8183 non-null    object 
 16  method              8183 non-null    object 
 17  rounds              8183 non-null    int64  
 18  time                8183 non-null    object 
 19  title_fight         8183 non-null    bool  
dtypes: bool(1), int64(9), object(10)
memory usage: 1.3+ MB
```

In [102]: df_fights['date'] = pd.to_datetime(df_fights['date'])

In [103]: df_fights['time'] = pd.to_datetime(df_fights['time'], format='%M:%S').dt.time

In [103]: df_fights.head()

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighter_1_sig_str | fighter_2_sig_str | fighter_1_td | fighter_2_td | fighter_1_sub | fighter_2_sub | result | winner | weight_class | method | rounds | time | title_fight |
|----------|---------------------|------------------|-------------|-----------------|-----------------|---------------------|---------------------|--------------------------|--------------------------|---------------------|---------------------|----------------------|----------------------|---------------|---------------|---------------------|---------------|---------------|-------------|--------------------|
| 0 | Arman Tsarukyan | Dan Hooker | 2025-11-22 | Doha, Qatar | Marc Goddard | 0 | 0 | | | | | | | | | | | | | |
| 1 | Ian Machado Garry | Belal Muhammad | 2025-11-22 | Doha, Qatar | Rich Mitchell | 0 | 0 | | | | | | | | | | | | | |
| 2 | Volkan Oezdemir | Alonzo Menifield | 2025-11-22 | Doha, Qatar | Marc Goddard | 1 | 0 | | | | | | | | | | | | | |
| 3 | Myktybek Orolbai | Jack Hermansson | 2025-11-22 | Doha, Qatar | Daniel Movahedi | 1 | 0 | | | | | | | | | | | | | |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | 2025-11-22 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | | | | | | | | | | | | | |

In [103...]: df_fights.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 8183 entries, 0 to 8207
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        8183 non-null    object  
 1   fighter_2        8183 non-null    object  
 2   date              8183 non-null    datetime64[ns]
 3   location          8183 non-null    object  
 4   referee            8183 non-null    object  
 5   fighter_1_kd      8183 non-null    int64  
 6   fighter_2_kd      8183 non-null    int64  
 7   fighter_1_sig_str 8183 non-null    int64  
 8   fighter_2_sig_str 8183 non-null    int64  
 9   fighter_1_td      8183 non-null    int64  
 10  fighter_2_td      8183 non-null    int64  
 11  fighter_1_sub     8183 non-null    int64  
 12  fighter_2_sub     8183 non-null    int64  
 13  result             8183 non-null    object  
 14  winner             8183 non-null    object  
 15  weight_class       8183 non-null    object  
 16  method              8183 non-null    object  
 17  rounds              8183 non-null    int64  
 18  time                8183 non-null    object  
 19  title_fight        8183 non-null    bool  
dtypes: bool(1), datetime64[ns](1), int64(9), object(9)
memory usage: 1.3+ MB
```

In [103...]: def print_unique_value(df):
for col in df:
 if df[col].dtype == "object":
 print(col, ":", df[col].unique())

In [103...]: df_fights.result.unique()

Out[103...]: array(['win', 'nc', 'draw\n\n\nndraw', 'nc\n\n\nnnc'], dtype=object)

In [103...]: df_fights['result'] = (
 df_fights['result']
 .str.lower()
 .str.replace(r'\s+', '', regex=True) # remove all whitespace/newlines
 .str.replace(r'(draw)+', 'draw', regex=True)
 .str.replace(r'(nc)+', 'nc', regex=True)
)

In [103...]: df_fights.result.unique()

Out[103...]: array(['win', 'nc', 'draw'], dtype=object)

In []: print_unique_value(df_fights)

In [103...]: df_fights[df_fights['result']=='nc']

Out[103...]

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fight |
|------|----------------------|------------------|-------------|--|------------------|---------------------|---------------------|--------------|
| 53 | Tom Aspinall | Ciryl Gane | 2025-10-25 | Abu Dhabi, Abu Dhabi, United Arab Emirates | Jason Herzog | 0 | 0 | 0 |
| 127 | Zach Reese | Sedriques Dumas | 2025-09-13 | San Antonio, Texas, USA | Mike Beltran | 0 | 0 | 0 |
| 259 | Paul Craig | Rodolfo Bellato | 2025-06-14 | Atlanta, Georgia, USA | Kevin MacDonald | 0 | 0 | 0 |
| 730 | Abdul Razak Alhassan | Cody Brundage | 2024-07-13 | Denver, Colorado, USA | Dan Miragliotta | 0 | 0 | 0 |
| 900 | Bryan Battle | Ange Loosa | 2024-03-16 | Las Vegas, Nevada, USA | Mike Beltran | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 7401 | Karo Parisyan | Dong Hyun Kim | 2009-01-31 | Las Vegas, Nevada, USA | Josh Rosenthal | 0 | 0 | 0 |
| 7707 | Gray Maynard | Rob Emerson | 2007-06-23 | Las Vegas, Nevada, USA | Steve Mazzagatti | 0 | 0 | 0 |
| 7970 | Alessio Sakara | Ron Faircloth | 2005-10-07 | Uncasville, Connecticut, USA | Herb Dean | 1 | 0 | 0 |
| 8143 | Benji Radach | Steve Berger | 2002-05-10 | Bossier City, Louisiana, USA | John McCarthy | 1 | 0 | 0 |
| 8198 | Bobby Hoffman | Mark Robinson | 2001-02-23 | Atlantic City, New Jersey, USA | John McCarthy | 1 | 0 | 0 |

88 rows × 20 columns



In [103...]

df_fights.shape[0] == df_fights[df_fights['result']=='nc'].shape[0] + df_fights[df_

Out[103...]

True

In [104...]

df_fights[df_fights['method']=="Overturned"]

Out[104...]

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd |
|------|---------------------|---------------------|-------------|------------------------------|-----------------|---------------------|---------------------|
| 257 | Mansur Abdul-Malik | Cody Brundage | 2025-06-14 | Atlanta, Georgia, USA | Brent McKeehan | 0 | 0 |
| 1058 | Trey Ogden | Nikolas Motta | 2023-11-18 | Las Vegas, Nevada, USA | Mike Beltran | 0 | 0 |
| 1122 | Dan Argueta | Miles Johns | 2023-09-23 | Las Vegas, Nevada, USA | Kerry Hatley | 0 | 0 |
| 1135 | Edgar Chairez | Daniel Lacerda | 2023-09-16 | Las Vegas, Nevada, USA | Chris Tognoni | 0 | 0 |
| 1299 | Dan Argueta | Ronnie Lawrence | 2023-06-17 | Las Vegas, Nevada, USA | Keith Peterson | 0 | 0 |
| 1731 | Don'Tale Mayes | Hamdy Abdelwahab | 2022-07-30 | Dallas, Texas, USA | Kerry Hatley | 0 | 1 |
| 2166 | Kevin Holland | Kyle Daukaus | 2021-10-02 | Las Vegas, Nevada, USA | Dan Miragliotta | 0 | 0 |
| 2359 | Matt Schnell | Rogerio Bontorin | 2021-05-15 | Houston, Texas, USA | Kerry Hatley | 0 | 0 |
| 2692 | Donald Cerrone | Niko Price | 2020-09-19 | Las Vegas, Nevada, USA | Jason Herzog | 0 | 0 |
| 3095 | Greg Hardy | Ben Sosoli | 2019-10-18 | Boston, Massachusetts, USA | Bryan Miner | 0 | 0 |
| 3258 | Mike Rodriguez | John Allan | 2019-07-13 | Sacramento, California, USA | Mike Beltran | 0 | 0 |
| 3528 | Andrei Arlovski | Walt Harris | 2018-12-29 | Los Angeles, California, USA | Mike Beltran | 0 | 0 |
| 3802 | CM Punk | Mike Jackson | 2018-06-09 | Chicago, Illinois, USA | Kevin MacDonald | 0 | 0 |
| 4002 | Khalil Rountree Jr. | Michal Oleksiejczuk | 2017-12-30 | Las Vegas, Nevada, USA | Herb Dean | 0 | 0 |
| 4424 | Adam Milstead | Curtis Blaydes | 2017-02-04 | Houston, Texas, USA | Kerry Hatley | 0 | 0 |
| 4537 | Francimmar Barroso | Darren Stewart | 2016-11-19 | Sao Paulo, Sao Paulo, Brazil | Eduardo Herdy | 0 | 0 |
| 4713 | Brock Lesnar | Mark Hunt | 2016-07-09 | Las Vegas, Nevada, USA | Dan Miragliotta | 0 | 0 |
| 5374 | Anderson Silva | Nick Diaz | 2015-01-31 | Las Vegas, Nevada, USA | John McCarthy | 0 | 0 |

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd |
|------|-------------------|------------------|-------------|-----------------------------------|----------------|---------------------|---------------------|
| 5413 | Hector Lombard | Joshua Burkman | 2015-01-03 | Las Vegas, Nevada, USA | John McCarthy | 0 | 0 |
| 5633 | Mike de la Torre | Brian Ortega | 2014-07-26 | San Jose, California, USA | John McCarthy | 0 | 0 |
| 5669 | Robert Drysdale | Keith Berish | 2014-07-06 | Las Vegas, Nevada, USA | Mark Smith | 0 | 0 |
| 5846 | Phil Harris | Louis Gaudinot | 2014-03-08 | London, England, United Kingdom | Grant Waterman | 0 | 0 |
| 5896 | Chico Camus | Yaotzin Meza | 2014-01-25 | Chicago, Illinois, USA | Otto Torriero | 0 | 0 |
| 5930 | Dennis Siver | Manvel Gamburyan | 2013-12-28 | Las Vegas, Nevada, USA | Herb Dean | 0 | 0 |
| 5999 | Yves Edwards | Yancy Medeiros | 2013-11-06 | Fort Campbell, Kentucky, USA | John McCarthy | 0 | 1 |
| 6021 | Sarah Kaufman | Jessica Eye | 2013-10-19 | Houston, Texas, USA | Jacob Montalvo | 0 | 0 |
| 6185 | Jim Miller | Pat Healy | 2013-04-27 | Newark, New Jersey, USA | Herb Dean | 0 | 0 |
| 6250 | Alex Caceres | Kyung Ho Kang | 2013-03-02 | Saitama, Japan | Herb Dean | 0 | 0 |
| 6269 | Che Mills | Matthew Riddle | 2013-02-16 | London, England, United Kingdom | Kevin Mulhall | 0 | 0 |
| 6337 | Igor Pokrajac | Joey Beltran | 2012-12-14 | Gold Coast, Queensland, Australia | John Sharp | 0 | 0 |
| 6366 | Thiago Silva | Stanislav Nedkov | 2012-11-10 | Macau, China | Steve Perceval | 0 | 1 |
| 6421 | Jake Shields | Ed Herman | 2012-08-11 | Denver, Colorado, USA | Adam Martinez | 0 | 0 |
| 6443 | Chris Clements | Matthew Riddle | 2012-07-21 | Calgary, Alberta, Canada | Josh Rosenthal | 0 | 0 |
| 6700 | Mackens Semerzier | Robert Peralta | 2011-11-12 | Anaheim, California, USA | John McCarthy | 0 | 2 |
| 6822 | Nik Lentz | Charles Oliveira | 2011-06-26 | Pittsburgh, Pennsylvania, USA | Chip Snider | 0 | 1 |

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd |
|-------------|------------------|------------------|-------------|--------------------------------|------------------|---------------------|---------------------|
| 6941 | Brandon Vera | Thiago Silva | 2011-01-01 | Las Vegas, Nevada, USA | Steve Mazzagatti | 0 | 0 |
| 7401 | Karo Parisyan | Dong Hyun Kim | 2009-01-31 | Las Vegas, Nevada, USA | Josh Rosenthal | 0 | 0 |
| 8143 | Benji Radach | Steve Berger | 2002-05-10 | Bossier City, Louisiana, USA | John McCarthy | 1 | 0 |
| 8198 | Bobby Hoffman | Mark Robinson | 2001-02-23 | Atlantic City, New Jersey, USA | John McCarthy | 1 | 0 |

In [104...]

```
# removing overturned and nc fights
df_fights = df_fights[df_fights['result']!='nc']
```

In [104...]

```
df_fights = df_fights[df_fights['method']!='Overturned']
```

In [104...]

```
df_fights[df_fights['winner']=='-'].shape[0]
```

Out[104...]

```
0
```

In [104...]

```
df_fights[df_fights['method']=='CNC'].shape[0]
```

Out[104...]

```
0
```

In [104...]

```
df_fights.shape
```

Out[104...]

```
(8094, 20)
```

In [104...]

```
df_fights.head()
```

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighter_1_td |
|----------|---------------------|------------------|-------------|-----------------|-----------------|---------------------|---------------------|---------------------|
| 0 | Arman Tsarukyan | Dan Hooker | 2025-11-22 | Doha, Qatar | Marc Goddard | 0 | 0 | 0 |
| 1 | Ian Machado Garry | Belal Muhammad | 2025-11-22 | Doha, Qatar | Rich Mitchell | 0 | 0 | 0 |
| 2 | Volkan Oezdemir | Alonzo Menfield | 2025-11-22 | Doha, Qatar | Marc Goddard | 1 | 0 | 0 |
| 3 | Myktybek Orolbai | Jack Hermansson | 2025-11-22 | Doha, Qatar | Daniel Movahedi | 1 | 0 | 0 |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | 2025-11-22 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | 0 |



```
In [104... df_fights[df_fights['fighter_1']==df_fights['winner']].shape
```

Out[104... (8040, 20)

```
In [104... df_fights.shape
```

Out[104... (8094, 20)

```
In [104... df_fights.loc[(df_fights['fighter_1']==df_fights['winner'])].shape[0] == df_fights
```

Out[104... True

```
In [105... df_fights.result.unique()
```

Out[105... array(['win', 'draw'], dtype=object)

```
In [105... df_fights[df_fights['result']=='draw'].shape[0]
```

Out[105... 54

```
In [105... df_fights.drop(df_fights[df_fights['result']=='draw'].index, inplace=True)
```

```
In [105... df_fights[df_fights['result']=='draw'].shape[0]
```

Out[105... 0

```
In [105... df_fights.result.unique()
```

Out[105... array(['win'], dtype=object)

```
In [105... df_fights.shape[0] == df_fights[df_fights['result']=='win'].shape[0]
```

Out[105... True

In [105... df_fights.shape[0] == df_fights.loc[(df_fights['fighter_1'] == df_fights['winner'])].

Out[105... True

In [105... df_fights.to_csv("./dataset/cleaned/ufc_fights_cleaned.csv", index=False)

In [105... df_fights.head()

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighter_1_sig |
|----------|---------------------|------------------|-------------|-----------------|-----------------|---------------------|---------------------|----------------------|
| 0 | Arman Tsarukyan | Dan Hooker | 2025-11-22 | Doha, Qatar | Marc Goddard | 0 | 0 | 0 |
| 1 | Ian Machado Garry | Belal Muhammad | 2025-11-22 | Doha, Qatar | Rich Mitchell | 0 | 0 | 0 |
| 2 | Volkan Oezdemir | Alonzo Menifield | 2025-11-22 | Doha, Qatar | Marc Goddard | 1 | 0 | 0 |
| 3 | Myktybek Orolbai | Jack Hermansson | 2025-11-22 | Doha, Qatar | Daniel Movahedi | 1 | 0 | 0 |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | 2025-11-22 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | 0 |

In [105...

```
def create_balanced_fight_data(df_original: pd.DataFrame) -> pd.DataFrame:
    """
    Balances a fight-by-fight dataset by creating a symmetric row for every fight.
    This corrects for positional bias (e.g., fighter_1 always being the winner) and
    ensures a 50/50 target distribution for modeling.
    """
    df = df_original.copy()

    # -----
    # 1. Create the Target Variable
    # -----
    # Target = 1 if fighter_1 is the winner (fighter in the f1 slot wins), 0 otherwise
    df['target'] = (df['fighter_1'] == df['winner']).astype(int)

    # -----
    # 2. Define Columns to Swap
    # -----
    # Pairs of columns that belong to fighter_1 and fighter_2, including names and
    SWAP_PAIRS = [
        ('fighter_1', 'fighter_2'),
        ('fighter_1_kd', 'fighter_2_kd'),
        ('fighter_1_sig_str', 'fighter_2_sig_str'),
        ('fighter_1_td', 'fighter_2_td'),
        ('fighter_1_sub', 'fighter_2_sub')
    ]
```

```
]

# -----
# 3. Create the Flipped Data Frame
# -----
df_flipped = df.copy()

# A) Swap F1 and F2 columns
for col_f1, col_f2 in SWAP_PAIRS:
    temp = df_flipped[col_f1].copy()
    df_flipped[col_f1] = df_flipped[col_f2]
    df_flipped[col_f2] = temp

# B) Invert the Target Variable (1 becomes 0, 0 becomes 1)
df_flipped['target'] = 1 - df_flipped['target']

# -----
# 4. Concatenate and Finalize
# -----
# Combine the original and the flipped dataframes
df_final_ml = pd.concat([df, df_flipped], ignore_index=True)

# Drop the original outcome columns that are now redundant or replaced by 'target'
df_final_ml = df_final_ml.drop(columns=['winner', 'result'])

return df_final_ml
```

```
In [106...]: df_fights_final = create_balanced_fight_data(df_fights)
```

```
In [106...]: df_fights_final.shape
```

```
Out[106...]: (16080, 19)
```

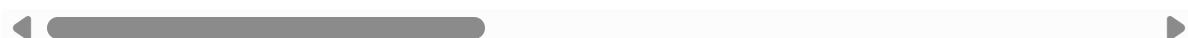
```
In [106...]: df_fights.shape
```

```
Out[106...]: (8040, 20)
```

```
In [106...]: df_fights_final.head()
```

Out[106...]

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighter_1_td |
|----------|---------------------|------------------|-------------|-----------------|-----------------|---------------------|---------------------|---------------------|
| 0 | Arman Tsarukyan | Dan Hooker | 2025-11-22 | Doha, Qatar | Marc Goddard | 0 | 0 | 0 |
| 1 | Ian Machado Garry | Belal Muhammad | 2025-11-22 | Doha, Qatar | Rich Mitchell | 0 | 0 | 0 |
| 2 | Volkan Oezdemir | Alonzo Menfield | 2025-11-22 | Doha, Qatar | Marc Goddard | 1 | 0 | 0 |
| 3 | Myktybek Orolbai | Jack Hermansson | 2025-11-22 | Doha, Qatar | Daniel Movahedi | 1 | 0 | 0 |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | 2025-11-22 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | 0 |



Merge the fighter and fight stat dataframes

In [106...]

```
df_f1 = df_fights_final.merge(
    df_fighters1,
    left_on="fighter_1",
    right_on="name",
    how="left",
    suffixes=("", "_f1")
)

df_final = df_f1.merge(
    df_fighters1,
    left_on="fighter_2",
    right_on="name",
    how="left",
    suffixes=("_f1", "_f2")
)
```

In [106...]

```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16080 entries, 0 to 16079
Data columns (total 59 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        16080 non-null   object  
 1   fighter_2        16080 non-null   object  
 2   date             16080 non-null   datetime64[ns]
 3   location          16080 non-null   object  
 4   referee           16080 non-null   object  
 5   fighter_1_kd     16080 non-null   int64  
 6   fighter_2_kd     16080 non-null   int64  
 7   fighter_1_sig_str 16080 non-null   int64  
 8   fighter_2_sig_str 16080 non-null   int64  
 9   fighter_1_td     16080 non-null   int64  
 10  fighter_2_td     16080 non-null   int64  
 11  fighter_1_sub    16080 non-null   int64  
 12  fighter_2_sub    16080 non-null   int64  
 13  weight_class     16080 non-null   object  
 14  method            16080 non-null   object  
 15  rounds            16080 non-null   int64  
 16  time              16080 non-null   object  
 17  title_fight       16080 non-null   bool   
 18  target             16080 non-null   int64  
 19  name_f1           16053 non-null   object  
 20  reach_f1          16053 non-null   float64 
 21  SLpM_f1           16053 non-null   float64 
 22  StrAcc_f1         16053 non-null   float64 
 23  SApM_f1           16053 non-null   float64 
 24  StrDef_f1         16053 non-null   float64 
 25  TDAvg_f1          16053 non-null   float64 
 26  TDAcc_f1          16053 non-null   float64 
 27  TDDef_f1          16053 non-null   float64 
 28  SubAvg_f1         16053 non-null   float64 
 29  height_cm_f1      16053 non-null   float64 
 30  weight_kg_f1      16053 non-null   float64 
 31  wins_f1            16053 non-null   float64 
 32  losses_f1          16053 non-null   float64 
 33  draws_f1           16053 non-null   float64 
 34  nc_f1              16053 non-null   float64 
 35  age_f1              16053 non-null   float64 
 36  total_fights_f1    16053 non-null   float64 
 37  win_rate_f1         16053 non-null   float64 
 38  stance_encoded_f1  16053 non-null   float64 
 39  name_f2              16053 non-null   object  
 40  reach_f2             16053 non-null   float64 
 41  SLpM_f2              16053 non-null   float64 
 42  StrAcc_f2             16053 non-null   float64 
 43  SApM_f2              16053 non-null   float64 
 44  StrDef_f2             16053 non-null   float64 
 45  TDAvg_f2             16053 non-null   float64 
 46  TDAcc_f2              16053 non-null   float64 
 47  TDDef_f2              16053 non-null   float64 
 48  SubAvg_f2             16053 non-null   float64 
 49  height_cm_f2          16053 non-null   float64 
 50  weight_kg_f2          16053 non-null   float64
```

```
51 wins_f2           16053 non-null  float64
52 losses_f2         16053 non-null  float64
53 draws_f2          16053 non-null  float64
54 nc_f2             16053 non-null  float64
55 age_f2            16053 non-null  float64
56 total_fights_f2   16053 non-null  float64
57 win_rate_f2       16053 non-null  float64
58 stance_encoded_f2 16053 non-null  float64
dtypes: bool(1), datetime64[ns](1), float64(38), int64(10), object(9)
memory usage: 7.1+ MB
```

```
In [106]: df_final1 = df_final.dropna()
```

```
In [106]: df_final1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16026 entries, 0 to 16079
Data columns (total 59 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        16026 non-null   object 
 1   fighter_2        16026 non-null   object 
 2   date             16026 non-null   datetime64[ns]
 3   location          16026 non-null   object 
 4   referee           16026 non-null   object 
 5   fighter_1_kd     16026 non-null   int64  
 6   fighter_2_kd     16026 non-null   int64  
 7   fighter_1_sig_str 16026 non-null   int64  
 8   fighter_2_sig_str 16026 non-null   int64  
 9   fighter_1_td     16026 non-null   int64  
 10  fighter_2_td     16026 non-null   int64  
 11  fighter_1_sub    16026 non-null   int64  
 12  fighter_2_sub    16026 non-null   int64  
 13  weight_class     16026 non-null   object 
 14  method            16026 non-null   object 
 15  rounds            16026 non-null   int64  
 16  time              16026 non-null   object 
 17  title_fight       16026 non-null   bool   
 18  target            16026 non-null   int64  
 19  name_f1           16026 non-null   object 
 20  reach_f1          16026 non-null   float64
 21  SLpM_f1           16026 non-null   float64
 22  StrAcc_f1         16026 non-null   float64
 23  SApM_f1           16026 non-null   float64
 24  StrDef_f1         16026 non-null   float64
 25  TDAvg_f1          16026 non-null   float64
 26  TDAcc_f1          16026 non-null   float64
 27  TDDef_f1          16026 non-null   float64
 28  SubAvg_f1         16026 non-null   float64
 29  height_cm_f1      16026 non-null   float64
 30  weight_kg_f1      16026 non-null   float64
 31  wins_f1            16026 non-null   float64
 32  losses_f1          16026 non-null   float64
 33  draws_f1           16026 non-null   float64
 34  nc_f1              16026 non-null   float64
 35  age_f1              16026 non-null   float64
 36  total_fights_f1    16026 non-null   float64
 37  win_rate_f1         16026 non-null   float64
 38  stance_encoded_f1  16026 non-null   float64
 39  name_f2              16026 non-null   object 
 40  reach_f2             16026 non-null   float64
 41  SLpM_f2              16026 non-null   float64
 42  StrAcc_f2             16026 non-null   float64
 43  SApM_f2              16026 non-null   float64
 44  StrDef_f2             16026 non-null   float64
 45  TDAvg_f2             16026 non-null   float64
 46  TDAcc_f2              16026 non-null   float64
 47  TDDef_f2              16026 non-null   float64
 48  SubAvg_f2             16026 non-null   float64
 49  height_cm_f2          16026 non-null   float64
 50  weight_kg_f2          16026 non-null   float64
```

```

51 wins_f2           16026 non-null float64
52 losses_f2        16026 non-null float64
53 draws_f2         16026 non-null float64
54 nc_f2            16026 non-null float64
55 age_f2           16026 non-null float64
56 total_fights_f2  16026 non-null float64
57 win_rate_f2      16026 non-null float64
58 stance_encoded_f2 16026 non-null float64
dtypes: bool(1), datetime64[ns](1), float64(38), int64(10), object(9)
memory usage: 7.2+ MB

```

Create delta features

```
In [ ]: df_final1["delta_height"] = df_final1["height_cm_f1"] - df_final1["height_cm_f2"]
df_final1["delta_reach"] = df_final1["reach_f1"] - df_final1["reach_f2"]
df_final1["delta_age"] = df_final1["age_f1"] - df_final1["age_f2"]
df_final1["delta_win_rate"] = df_final1["win_rate_f1"] - df_final1["win_ra"]
df_final1["delta_SLPm"] = df_final1["SLPm_f1"] - df_final1["SLPm_f2"]
df_final1["delta_StrAcc"] = df_final1["StrAcc_f1"] - df_final1["StrAcc_f2"]
df_final1["delta_SApM"] = df_final1["SApM_f1"] - df_final1["SApM_f2"]
df_final1["delta_StrDef"] = df_final1["StrDef_f1"] - df_final1["StrDef_f2"]
df_final1["delta_TDAvg"] = df_final1["TDAvg_f1"] - df_final1["TDAvg_f2"]
df_final1["delta_TDAcc"] = df_final1["TDAcc_f1"] - df_final1["TDAcc_f2"]
df_final1["delta_TDDef"] = df_final1["TDDef_f1"] - df_final1["TDDef_f2"]
df_final1["delta_SubAvg"] = df_final1["SubAvg_f1"] - df_final1["SubAvg_f2"]
```

```
In [106...]: df_final1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16026 entries, 0 to 16079
Data columns (total 71 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        16026 non-null   object  
 1   fighter_2        16026 non-null   object  
 2   date             16026 non-null   datetime64[ns]
 3   location          16026 non-null   object  
 4   referee           16026 non-null   object  
 5   fighter_1_kd     16026 non-null   int64  
 6   fighter_2_kd     16026 non-null   int64  
 7   fighter_1_sig_str 16026 non-null   int64  
 8   fighter_2_sig_str 16026 non-null   int64  
 9   fighter_1_td     16026 non-null   int64  
 10  fighter_2_td     16026 non-null   int64  
 11  fighter_1_sub    16026 non-null   int64  
 12  fighter_2_sub    16026 non-null   int64  
 13  weight_class     16026 non-null   object  
 14  method            16026 non-null   object  
 15  rounds            16026 non-null   int64  
 16  time              16026 non-null   object  
 17  title_fight       16026 non-null   bool   
 18  target             16026 non-null   int64  
 19  name_f1           16026 non-null   object  
 20  reach_f1          16026 non-null   float64 
 21  SLpM_f1           16026 non-null   float64 
 22  StrAcc_f1         16026 non-null   float64 
 23  SApM_f1           16026 non-null   float64 
 24  StrDef_f1         16026 non-null   float64 
 25  TDAvg_f1          16026 non-null   float64 
 26  TDAcc_f1          16026 non-null   float64 
 27  TDDef_f1          16026 non-null   float64 
 28  SubAvg_f1         16026 non-null   float64 
 29  height_cm_f1      16026 non-null   float64 
 30  weight_kg_f1      16026 non-null   float64 
 31  wins_f1            16026 non-null   float64 
 32  losses_f1          16026 non-null   float64 
 33  draws_f1           16026 non-null   float64 
 34  nc_f1              16026 non-null   float64 
 35  age_f1              16026 non-null   float64 
 36  total_fights_f1    16026 non-null   float64 
 37  win_rate_f1         16026 non-null   float64 
 38  stance_encoded_f1  16026 non-null   float64 
 39  name_f2              16026 non-null   object  
 40  reach_f2             16026 non-null   float64 
 41  SLpM_f2              16026 non-null   float64 
 42  StrAcc_f2             16026 non-null   float64 
 43  SApM_f2              16026 non-null   float64 
 44  StrDef_f2             16026 non-null   float64 
 45  TDAvg_f2             16026 non-null   float64 
 46  TDAcc_f2              16026 non-null   float64 
 47  TDDef_f2              16026 non-null   float64 
 48  SubAvg_f2             16026 non-null   float64 
 49  height_cm_f2          16026 non-null   float64 
 50  weight_kg_f2          16026 non-null   float64
```

```

51 wins_f2           16026 non-null float64
52 losses_f2        16026 non-null float64
53 draws_f2          16026 non-null float64
54 nc_f2             16026 non-null float64
55 age_f2            16026 non-null float64
56 total_fights_f2   16026 non-null float64
57 win_rate_f2       16026 non-null float64
58 stance_encoded_f2 16026 non-null float64
59 delta_height      16026 non-null float64
60 delta_reach        16026 non-null float64
61 delta_age          16026 non-null float64
62 delta_win_rate    16026 non-null float64
63 delta_SLPM         16026 non-null float64
64 delta_StrAcc       16026 non-null float64
65 delta_SApM         16026 non-null float64
66 delta_StrDef       16026 non-null float64
67 delta_TDAvg        16026 non-null float64
68 delta_TDAcc        16026 non-null float64
69 delta_TDDef        16026 non-null float64
70 delta_SubAvg       16026 non-null float64
dtypes: bool(1), datetime64[ns](1), float64(50), int64(10), object(9)
memory usage: 8.7+ MB

```

In [107... df_final1.head()

| | fighter_1 | fighter_2 | date | location | referee | fighter_1_kd | fighter_2_kd | fighter_1_s |
|----------|---------------------|------------------|-------------|-----------------|-----------------|---------------------|---------------------|--------------------|
| 0 | Arman Tsarukyan | Dan Hooker | 2025-11-22 | Doha, Qatar | Marc Goddard | 0 | 0 | |
| 1 | Ian Machado Garry | Belal Muhammad | 2025-11-22 | Doha, Qatar | Rich Mitchell | 0 | 0 | |
| 2 | Volkan Oezdemir | Alonzo Menifield | 2025-11-22 | Doha, Qatar | Marc Goddard | 1 | 0 | |
| 3 | Myktybek Orolbai | Jack Hermansson | 2025-11-22 | Doha, Qatar | Daniel Movahedi | 1 | 0 | |
| 4 | Waldo Cortes Acosta | Shamil Gaziev | 2025-11-22 | Doha, Qatar | Lukasz Bosacki | 2 | 0 | |

5 rows × 71 columns

In [107... df_final1.to_csv("./dataset/cleaned/ufc_full_cleaned.csv")

```

In [107... numeric_df = df_final1.select_dtypes(include=['float64', 'int64'])

# Compute correlation matrix
corr = numeric_df.corr()

# Plot heatmap (no manual colors used)

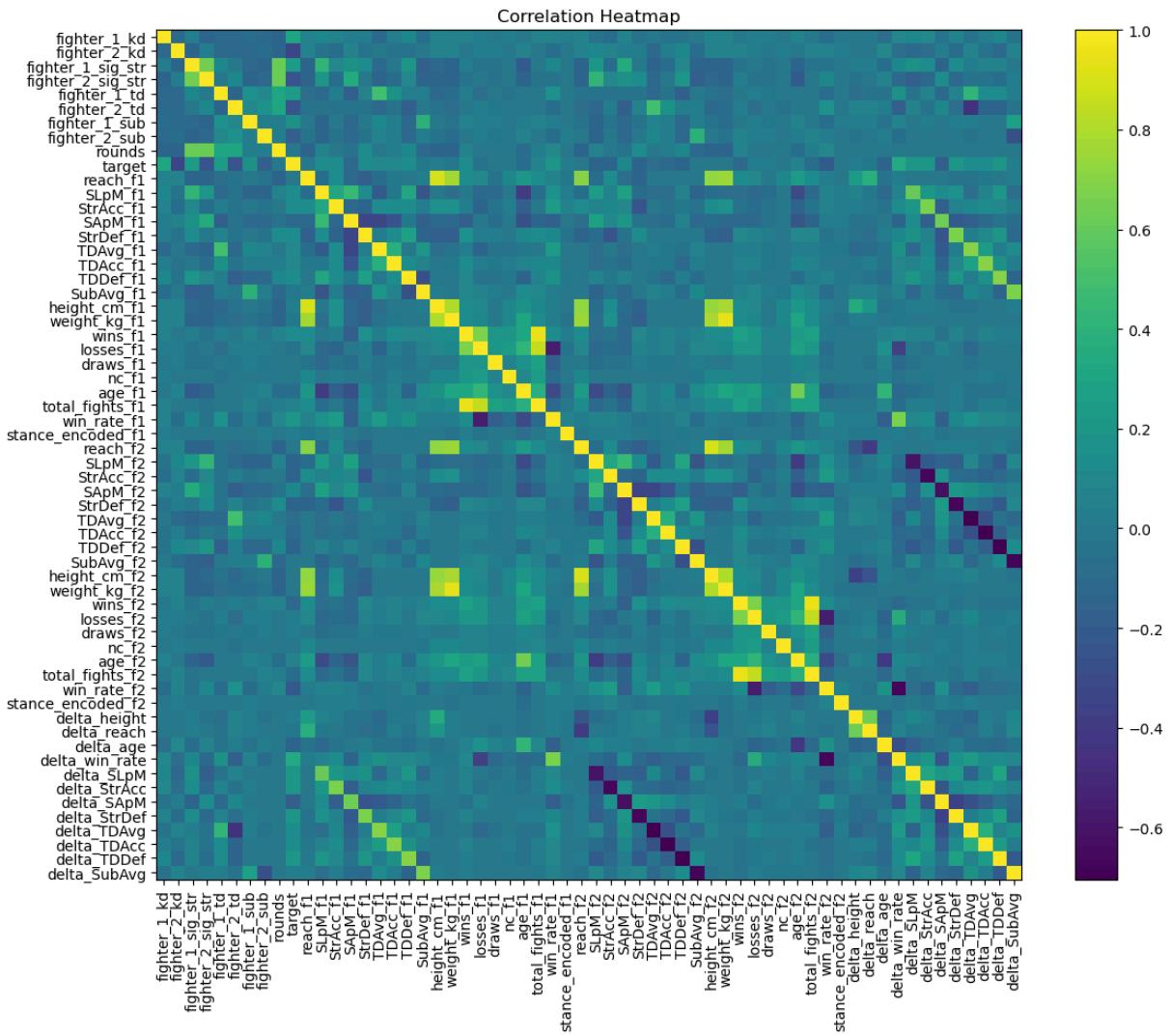
```

```

plt.figure(figsize=(12, 10))
plt.imshow(corr, aspect='auto')
plt.colorbar()
plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
plt.yticks(range(len(corr.columns)), corr.columns)
plt.title("Correlation Heatmap")

plt.tight_layout()
plt.show()

```



In [107]: df_final1.shape[0]

Out[107]: 16026

In [107]: df_final1.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 16026 entries, 0 to 16079
Data columns (total 71 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   fighter_1        16026 non-null   object  
 1   fighter_2        16026 non-null   object  
 2   date             16026 non-null   datetime64[ns]
 3   location          16026 non-null   object  
 4   referee           16026 non-null   object  
 5   fighter_1_kd     16026 non-null   int64  
 6   fighter_2_kd     16026 non-null   int64  
 7   fighter_1_sig_str 16026 non-null   int64  
 8   fighter_2_sig_str 16026 non-null   int64  
 9   fighter_1_td     16026 non-null   int64  
 10  fighter_2_td     16026 non-null   int64  
 11  fighter_1_sub    16026 non-null   int64  
 12  fighter_2_sub    16026 non-null   int64  
 13  weight_class     16026 non-null   object  
 14  method            16026 non-null   object  
 15  rounds            16026 non-null   int64  
 16  time              16026 non-null   object  
 17  title_fight       16026 non-null   bool   
 18  target             16026 non-null   int64  
 19  name_f1           16026 non-null   object  
 20  reach_f1          16026 non-null   float64 
 21  SLpM_f1           16026 non-null   float64 
 22  StrAcc_f1         16026 non-null   float64 
 23  SApM_f1           16026 non-null   float64 
 24  StrDef_f1         16026 non-null   float64 
 25  TDAvg_f1          16026 non-null   float64 
 26  TDAcc_f1          16026 non-null   float64 
 27  TDDef_f1          16026 non-null   float64 
 28  SubAvg_f1         16026 non-null   float64 
 29  height_cm_f1      16026 non-null   float64 
 30  weight_kg_f1      16026 non-null   float64 
 31  wins_f1            16026 non-null   float64 
 32  losses_f1          16026 non-null   float64 
 33  draws_f1           16026 non-null   float64 
 34  nc_f1              16026 non-null   float64 
 35  age_f1              16026 non-null   float64 
 36  total_fights_f1    16026 non-null   float64 
 37  win_rate_f1         16026 non-null   float64 
 38  stance_encoded_f1  16026 non-null   float64 
 39  name_f2              16026 non-null   object  
 40  reach_f2             16026 non-null   float64 
 41  SLpM_f2              16026 non-null   float64 
 42  StrAcc_f2             16026 non-null   float64 
 43  SApM_f2              16026 non-null   float64 
 44  StrDef_f2             16026 non-null   float64 
 45  TDAvg_f2             16026 non-null   float64 
 46  TDAcc_f2              16026 non-null   float64 
 47  TDDef_f2              16026 non-null   float64 
 48  SubAvg_f2             16026 non-null   float64 
 49  height_cm_f2          16026 non-null   float64 
 50  weight_kg_f2          16026 non-null   float64
```

```
51 wins_f2           16026 non-null  float64
52 losses_f2        16026 non-null  float64
53 draws_f2         16026 non-null  float64
54 nc_f2            16026 non-null  float64
55 age_f2           16026 non-null  float64
56 total_fights_f2  16026 non-null  float64
57 win_rate_f2      16026 non-null  float64
58 stance_encoded_f2 16026 non-null  float64
59 delta_height     16026 non-null  float64
60 delta_reach      16026 non-null  float64
61 delta_age         16026 non-null  float64
62 delta_win_rate   16026 non-null  float64
63 delta_SLPM        16026 non-null  float64
64 delta_StrAcc     16026 non-null  float64
65 delta_SApM        16026 non-null  float64
66 delta_StrDef     16026 non-null  float64
67 delta_TDAvg       16026 non-null  float64
68 delta_TDAcc       16026 non-null  float64
69 delta_TDDef       16026 non-null  float64
70 delta_SubAvg      16026 non-null  float64
dtypes: bool(1), datetime64[ns](1), float64(50), int64(10), object(9)
memory usage: 8.7+ MB
```

Feature Selection

```
In [107]: df_ml_final = df_final[['target','delta_height','delta_reach','delta_age','delta_w  
          'delta_SApM','delta_StrDef','delta_TDAvg','delta_TDAcc','d
```

```
In [107]: df_ml_final.head()
```

| | target | delta_height | delta_reach | delta_age | delta_win_rate | delta_SLPM | delta_StrAcc | delta_GoalAcc |
|---|--------|--------------|-------------|-----------|----------------|------------|--------------|---------------|
| 0 | 1 | -12.70 | -7.62 | -6.0 | 0.235967 | -1.04 | 0.02 | 0.00 |
| 1 | 1 | 10.16 | 5.08 | -9.0 | 0.116858 | 0.35 | 0.11 | 0.00 |
| 2 | 1 | 5.08 | -2.54 | -2.0 | 0.015805 | 1.22 | -0.05 | 0.00 |
| 3 | 1 | -7.62 | -7.62 | -10.0 | 0.127451 | -1.91 | 0.03 | 0.00 |
| 4 | 1 | 0.00 | 0.00 | -1.0 | 0.013889 | 2.93 | 0.01 | 0.00 |

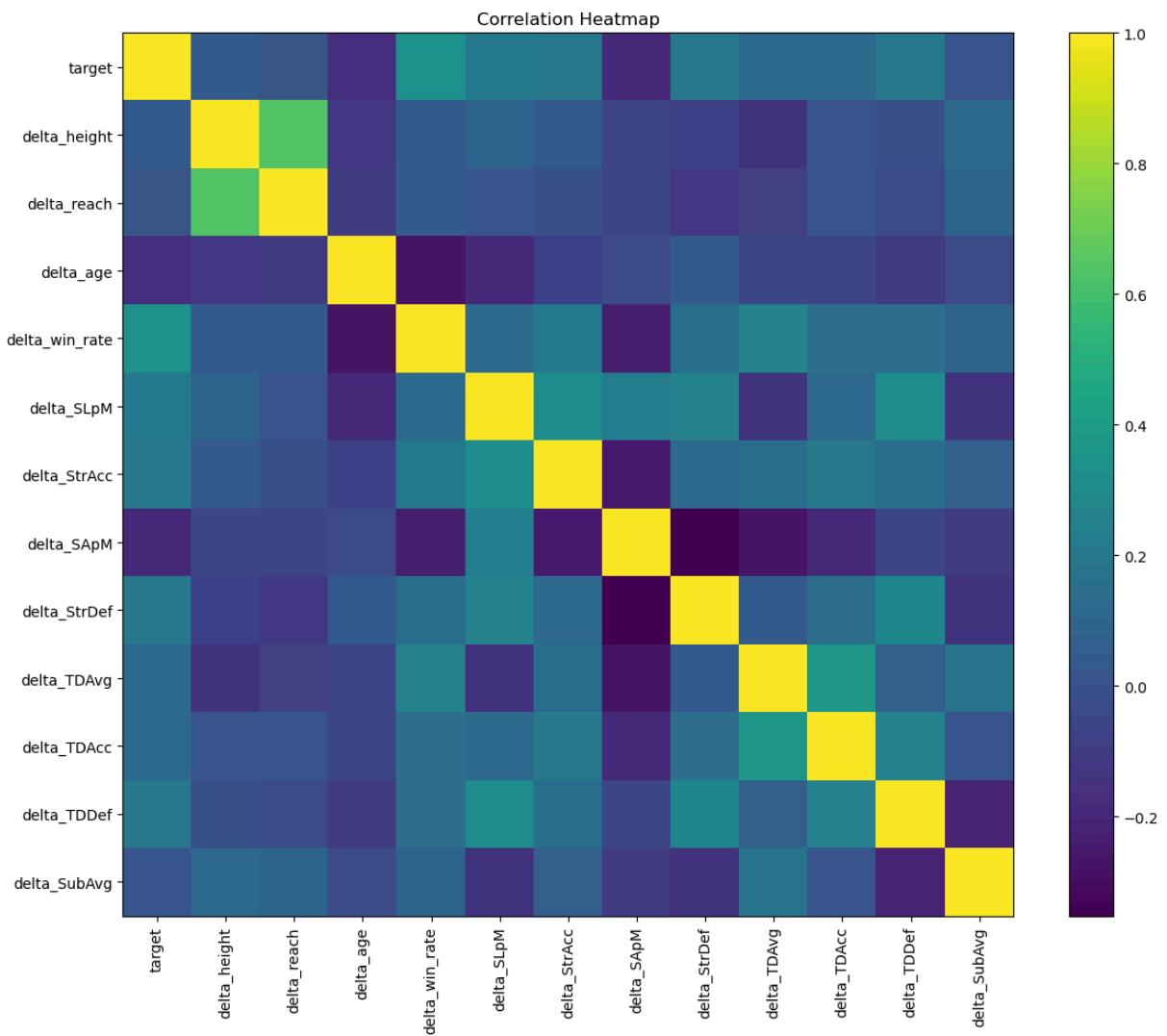
```
In [112...]: numeric_df = df_ml_final.select_dtypes(include=['float64', 'int64'])

# Compute correlation matrix
corr = numeric_df.corr()

# Plot heatmap (no manual colors used)
plt.figure(figsize=(12, 10))
plt.imshow(corr, aspect='auto')
plt.colorbar()
plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
plt.yticks(range(len(corr.columns)), corr.columns)
```

```
plt.title("Correlation Heatmap")

plt.tight_layout()
plt.show()
```



In [107...]:

```
x = df_ml_final.drop(columns='target')
y = df_ml_final[['target']]
```

In [107...]:

```
from sklearn.model_selection import train_test_split
```

In [107...]:

```
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size=0.2, shuffle=True)
```

In [108...]:

```
from sklearn.preprocessing import StandardScaler
```

In [108...]:

```
scaler = StandardScaler()
```

In [108...]:

```
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

ML Model Building

```
In [108...]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
```

1) Random Forest Classifier

```
In [108...]: rf_model = RandomForestClassifier(n_estimators=200,
                                         random_state=42,
                                         class_weight='balanced',
                                         max_depth=15,
                                         min_samples_leaf=5,
                                         n_jobs=-1) # Use all processors
```

```
In [108...]: rf_model.fit(x_train_scaled, y_train)
```

C:\Anaconda3\Lib\site-packages\sklearn\base.py:1389: DataConversionWarning: A column -vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
 return fit_method(estimator, *args, **kwargs)

```
Out[108...]: RandomForestClassifier
```

```
RandomForestClassifier(class_weight='balanced', max_depth=15,
                      min_samples_leaf=5, n_estimators=200, n_jobs=-1,
                      random_state=42)
```

```
In [108...]: y_pred = rf_model.predict(x_test_scaled)
```

```
In [108...]: accuracy = accuracy_score(y_test, y_pred)
```

```
In [108...]: accuracy
```

```
Out[108...]: 0.6818465377417342
```

```
In [108...]: report = classification_report(y_test, y_pred, digits=4)

print("\n" + "="*50)
print("MODEL EVALUATION RESULTS (Random Forest)")
print("="*50)
print(f"Test Set Accuracy: {accuracy:.4f}")
print("\nClassification Report (Test Set):\n")
print(report)
```

```
=====
MODEL EVALUATION RESULTS (Random Forest)
=====
Test Set Accuracy: 0.6818
```

Classification Report (Test Set):

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.6753 | 0.6906 | 0.6828 | 1590 |
| 1 | 0.6886 | 0.6733 | 0.6809 | 1616 |
| accuracy | | | 0.6818 | 3206 |
| macro avg | 0.6819 | 0.6819 | 0.6818 | 3206 |
| weighted avg | 0.6820 | 0.6818 | 0.6818 | 3206 |

```
In [109...]: feature_importances = pd.Series(rf_model.feature_importances_, index=x.columns).sort_values(ascending=False)

print("-" * 50)
print("Top 10 Feature Importances:")
print("-" * 50)
print(feature_importances.to_markdown(numalign='left', stralign='left'))
```

Top 10 Feature Importances:

| | 0 |
|----------------|-----------|
| delta_win_rate | 0.218699 |
| delta_SLPM | 0.125585 |
| delta_SApM | 0.0995426 |
| delta_TDDef | 0.08926 |
| delta_StrAcc | 0.0761618 |
| delta_TDAvg | 0.0742875 |
| delta_StrDef | 0.0708474 |
| delta_TDAcc | 0.0676787 |
| delta_age | 0.0561872 |
| delta_SubAvg | 0.0520331 |

2) xgBoost

```
In [109...]: from xgboost import XGBClassifier
```

```
In [109... xgb_model = XGBClassifier(
    objective='binary:logistic',
    n_estimators=300,
    learning_rate=0.05, # Using a smaller Learning rate and more estimators for
    max_depth=5,
    use_label_encoder=False,
    eval_metric='logloss',
    random_state=42,
    n_jobs=-1
)
```

```
In [109... xgb_model.fit(x_train_scaled,y_train)
```

C:\Anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning: [08:49:02] WARNING: C:\actions-runner_work\xgboost\xgboost\src\learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

```
bst.update(dtrain, iteration=i, fobj=obj)
```

```
Out[109... ▾ XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds
= None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, feature_weights=None, gamma=None,
              grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.05, max_bin
= None,
```

```
In [109... xgb_test_preds = xgb_model.predict(x_test_scaled)
```

```
In [109... accuracy_xgb = accuracy_score(xgb_test_preds,y_test)
```

```
In [111... # Accuracy score for test set
accuracy_xgb
```

```
Out[111... 0.6899563318777293
```

```
In [111... # classification report
report = classification_report(y_test, xgb_test_preds, digits=4)

print("\n" + "*50)
print("MODEL EVALUATION RESULTS (xgBoost)")
print("*50)
print(f"Test Set Accuracy: {accuracy_xgb:.4f}")
print("\nClassification Report (Test Set):\n")
print(report)
```

```
=====
MODEL EVALUATION RESULTS (xgBoost)
=====
Test Set Accuracy: 0.6900
```

Classification Report (Test Set):

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.6865 | 0.6899 | 0.6882 | 1590 |
| 1 | 0.6934 | 0.6900 | 0.6917 | 1616 |
| accuracy | | | 0.6900 | 3206 |
| macro avg | 0.6899 | 0.6900 | 0.6899 | 3206 |
| weighted avg | 0.6900 | 0.6900 | 0.6900 | 3206 |

Model Testing

In [111...]

```
def predict_Winner(f1, f2):

    f1_stats = df_fighters1[df_fighters1['name'] == f1]
    f2_stats = df_fighters1[df_fighters1['name'] == f2]

    if f1_stats.empty:
        return f"Fighter not found: {f1}"
    if f2_stats.empty:
        return f"Fighter not found: {f2}"

    f1_stats = f1_stats.iloc[0]
    f2_stats = f2_stats.iloc[0]

    df_pred = pd.DataFrame([
        'delta_height': f1_stats.height_cm - f2_stats.height_cm,
        'delta_reach': f1_stats.reach - f2_stats.reach,
        'delta_age': f1_stats.age - f2_stats.age,
        'delta_win_rate': f1_stats.win_rate - f2_stats.win_rate,
        'delta_SLPm': f1_stats.SLPm - f2_stats.SLPm,
        'delta_StrAcc': f1_stats.StrAcc - f2_stats.StrAcc,
        'delta_SApM': f1_stats.SApM - f2_stats.SApM,
        'delta_StrDef': f1_stats.StrDef - f2_stats.StrDef,
        'delta_TDAvg': f1_stats.TDAvg - f2_stats.TDAvg,
        'delta_TDAcc': f1_stats.TDAcc - f2_stats.TDAcc,
        'delta_TDDef': f1_stats.TDDef - f2_stats.TDDef,
        'delta_SubAvg': f1_stats.SubAvg - f2_stats.SubAvg
    ])

    df_pred_scaled = scaler.transform(df_pred)

    prob_f1_wins = rf_model.predict_proba(df_pred_scaled)[0][1]
```

```

prob_f2_wins = 1 - prob_f1_wins

predicted_winner = f1 if prob_f1_wins >= 0.5 else f2
confidence = max(prob_f1_wins, prob_f2_wins)

return {
    "fighter_1": f1,
    "fighter_2": f2,
    "predicted_winner": predicted_winner,
    "winner_probability": round(float(confidence), 4),
    f"{f1}_win_prob": round(float(prob_f1_wins), 4),
    f"{f2}_win_prob": round(float(prob_f2_wins), 4),
    "delta_features_used": df_pred.to_dict(orient="records")[0]
}

```

In [113... predict_Winner('Alexander Volkov', 'Jon Jones')

Out[113... {'fighter_1': 'Alexander Volkov',
'fighter_2': 'Jon Jones',
'predicted_winner': 'Jon Jones',
'winner_probability': 0.7584,
'Alexander Volkov_win_prob': 0.2416,
'Jon Jones_win_prob': 0.7584,
'delta_features_used': {'delta_height': 7.620000000000045,
'delta_reach': -10.160000000000025,
'delta_age': -1,
'delta_win_rate': -0.18551724137931036,
'delta_SLPM': 0.4000000000000036,
'delta_StrAcc': -0.0100000000000009,
'delta_SApM': 0.6199999999999997,
'delta_StrDef': -0.1099999999999999,
'delta_TDAvg': -1.319999999999998,
'delta_TDAcc': 0.2100000000000002,
'delta_TDDef': -0.2699999999999999,
'delta_SubAvg': -0.3}}

Neural Network

In [110... from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

In [110... nn_model = Sequential([
Dense(64, activation='relu', input_shape=(12,)), # Input Layer
Dense(32, activation='relu'), # Hidden Layer 1
Dense(16, activation='relu'), # Hidden Layer 2
Dense(1, activation='sigmoid') # Output Layer (Sigmoid for binary classification)
])
Compile the model
nn_model.compile(
optimizer='adam',

```

        loss='binary_crossentropy',
        metrics=['accuracy']
    )

```

C:\Anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:92: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)

```
In [110...]: print(nn_model.summary())
print("\nTraining Neural Network...")
```

Model: "sequential_1"

| Layer (type) | Output Shape | Param # |
|-----------------|--------------|---------|
| dense_4 (Dense) | (None, 64) | 832 |
| dense_5 (Dense) | (None, 32) | 2,080 |
| dense_6 (Dense) | (None, 16) | 528 |
| dense_7 (Dense) | (None, 1) | 17 |

Total params: 3,457 (13.50 KB)

Trainable params: 3,457 (13.50 KB)

Non-trainable params: 0 (0.00 B)

None

Training Neural Network...

```
In [110...]: history = nn_model.fit(
    x_train_scaled, y_train,
    epochs=20,
)
```

```

Epoch 1/20
401/401 ━━━━━━━━ 2s 1ms/step - accuracy: 0.6592 - loss: 0.6082
Epoch 2/20
401/401 ━━━━━━ 1s 1ms/step - accuracy: 0.6783 - loss: 0.5894
Epoch 3/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.6799 - loss: 0.5852
Epoch 4/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.6829 - loss: 0.5831
Epoch 5/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.6862 - loss: 0.5793
Epoch 6/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.6885 - loss: 0.5768
Epoch 7/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.6910 - loss: 0.5739
Epoch 8/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.6949 - loss: 0.5714
Epoch 9/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7012 - loss: 0.5684
Epoch 10/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.7010 - loss: 0.5654
Epoch 11/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7062 - loss: 0.5630
Epoch 12/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.7073 - loss: 0.5604
Epoch 13/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7090 - loss: 0.5575
Epoch 14/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.7095 - loss: 0.5558
Epoch 15/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7122 - loss: 0.5522
Epoch 16/20
401/401 ━━━━ 0s 1ms/step - accuracy: 0.7142 - loss: 0.5501
Epoch 17/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7156 - loss: 0.5494
Epoch 18/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7182 - loss: 0.5464
Epoch 19/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7187 - loss: 0.5429
Epoch 20/20
401/401 ━━━━ 1s 1ms/step - accuracy: 0.7171 - loss: 0.5407

```

```
In [110]: nn_prediction = nn_model.predict(x_test_scaled)
```

```
101/101 ━━━━━━ 0s 1ms/step
```

```
In [112]: y_pred_nn = (nn_prediction > 0.5).astype(int)
```

```
In [112]: nn_model.evaluate(x_test_scaled,y_test)
```

```
101/101 ━━━━━━ 1s 3ms/step - accuracy: 0.6694 - loss: 0.6028
```

```
Out[112]: [0.602819561958313, 0.6693699359893799]
```

```
In [ ]:
```