In [1]:
```
#Import pandas
import pandas as pd
```

In [2]:
```
#Load stats data in df
df = pd.read_csv("C:/Users/rohan/Downloads/UFC/new_stats.csv")
```

In [3]:
```
#First 30 rows
df.head(30)
```

Out[3]:

	fighter_id	sig_strikes_succ	takedowns_att	takedowns_succ	submiss_att	reversals	ctrl	knockdown
0	2.0	59.0	0.0	0.0	1.0	0.0	0	0.(
1	6.0	0.0	0.0	0.0	0.0	0.0	0	0.(
2	7.0	181.0	20.0	13.0	4.0	4.0	1866	0.(
3	9.0	29.0	1.0	0.0	1.0	0.0	166	0.(
4	11.0	218.0	21.0	5.0	0.0	3.0	720	0.(
5	12.0	425.0	36.0	7.0	1.0	0.0	533	0.(
6	13.0	66.0	12.0	2.0	0.0	2.0	393	0.(
7	16.0	738.0	45.0	19.0	3.0	2.0	1996	3.(
8	17.0	77.0	23.0	7.0	3.0	0.0	1069	0.(
9	18.0	52.0	2.0	2.0	5.0	0.0	122	0.(
10	23.0	166.0	1.0	0.0	1.0	0.0	121	1.(
11	24.0	25.0	18.0	1.0	2.0	0.0	388	0.(
12	25.0	9.0	0.0	0.0	0.0	0.0	0	0.(
13	26.0	238.0	12.0	4.0	7.0	1.0	376	1.(
14	28.0	28.0	1.0	0.0	1.0	0.0	24	0.(
15	30.0	719.0	35.0	5.0	4.0	2.0	1011	2.(
16	31.0	226.0	47.0	14.0	7.0	5.0	1896	1.(
17	33.0	186.0	6.0	1.0	0.0	1.0	163	3.(
18	34.0	22.0	5.0	3.0	0.0	0.0	251	0.(
19	35.0	92.0	3.0	1.0	0.0	0.0	1	0.(
20	36.0	36.0	1.0	0.0	2.0	3.0	96	0.(
21	37.0	323.0	7.0	1.0	3.0	1.0	792	2.(
22	38.0	7.0	1.0	1.0	0.0	0.0	137	0.(
23	40.0	71.0	3.0	2.0	0.0	0.0	274	0.(
24	41.0	140.0	1.0	0.0	2.0	1.0	102	0.(
25	42.0	373.0	16.0	3.0	0.0	0.0	345	1.(

	fighter_id	sig_strikes_succ	takedowns_att	takedowns_succ	submiss_att	reversals	ctrl	knockdown
26	43.0	135.0	9.0	6.0	0.0	0.0	612	0.(
27	48.0	18.0	7.0	2.0	4.0	2.0	406	0.(
28	54.0	462.0	40.0	12.0	6.0	5.0	2603	1.(
29	60.0	6.0	0.0	0.0	0.0	0.0	0	0.(

In [4]:
```
'''
Calculating the following:
TSR: Total strike succeeded ratio
SSR: Significant strike succeeded ratio
TKR: Total takedowns succeeded ratio
'''

df['TSR'] = df['total_strikes_succ'] / df['total_strikes_att']
df['SSR'] = df['sig_strikes_succ'] / df['sig_strikes_att']
df['TKR'] = df['takedowns_succ'] / df['takedowns_att']
```

In [5]:
```
#Dropping tables which are not useful

df = df.drop(['takedowns_succ','takedowns_att','total_strikes_succ',
              'total_strikes_att','sig_strikes_succ','sig_strikes_att'], axis=1)
```

In [6]:
```
df
```

Out[6]:

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR
0	2.0	1.0	0.0	0	0.0	0.632653	0.526786	NaN
1	6.0	0.0	0.0	0	0.0	0.000000	0.000000	NaN
2	7.0	4.0	4.0	1866	0.0	0.747646	0.613559	0.650000
3	9.0	1.0	0.0	166	0.0	0.591398	0.439394	0.000000
4	11.0	0.0	3.0	720	0.0	0.590551	0.470842	0.238095
...	...	...	...	...	...	...	...	...
2407	4099.0	3.0	0.0	804	0.0	0.613208	0.551136	0.571429
2408	4100.0	0.0	0.0	22	1.0	0.339806	0.336614	0.500000
2409	4102.0	1.0	0.0	738	2.0	0.538569	0.442681	0.233333
2410	4104.0	4.0	1.0	123	4.0	0.583490	0.444444	0.625000
2411	4106.0	0.0	0.0	110	0.0	0.459459	0.380645	NaN

2412 rows × 8 columns

In [7]:
```
df
```

Out[7]:

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR
0	2.0	1.0	0.0	0	0.0	0.632653	0.526786	NaN
1	6.0	0.0	0.0	0	0.0	0.000000	0.000000	NaN
2	7.0	4.0	4.0	1866	0.0	0.747646	0.613559	0.650000
3	9.0	1.0	0.0	166	0.0	0.591398	0.439394	0.000000
4	11.0	0.0	3.0	720	0.0	0.590551	0.470842	0.238095
...	...	...	...	...	...	...	...	...
2407	4099.0	3.0	0.0	804	0.0	0.613208	0.551136	0.571429
2408	4100.0	0.0	0.0	22	1.0	0.339806	0.336614	0.500000
2409	4102.0	1.0	0.0	738	2.0	0.538569	0.442681	0.233333
2410	4104.0	4.0	1.0	123	4.0	0.583490	0.444444	0.625000
2411	4106.0	0.0	0.0	110	0.0	0.459459	0.380645	NaN

2412 rows × 8 columns

In [8]:
```
#Loading fighter dataset in df1
df1 = pd.read_csv("C:/Users/rohan/Downloads/UFC/UFC_FIGHTER.csv")
```

In [9]:
```
#Merging stats df and fighter df1 using left join
df8 = pd.merge(df, df1[['fighter_id','reach_cm', 'wins', 'loss']], on='fighter_id', how
```

In [10]:
```
#Merging again for future slicing. Storing as copy with name and id as extra columns
df9 = pd.merge(df, df1[['fighter_id','first_name','nickname','last_name','reach_cm', 'w
```

In [11]:
```
df9
```

Out[11]:

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR	first_name	ni
0	2.0	1.0	0.0	0	0.0	0.632653	0.526786	NaN	Allan	
1	6.0	0.0	0.0	0	0.0	0.000000	0.000000	NaN	Igor	
2	7.0	4.0	4.0	1866	0.0	0.747646	0.613559	0.650000	Cat	
3	9.0	1.0	0.0	166	0.0	0.591398	0.439394	0.000000	James	Me
4	11.0	0.0	3.0	720	0.0	0.590551	0.470842	0.238095	Fares	
...	...	...	...	...	...	...	...	...	...	
2407	4099.0	3.0	0.0	804	0.0	0.613208	0.551136	0.571429	Papy	M
2408	4100.0	0.0	0.0	22	1.0	0.339806	0.336614	0.500000	Daichi	
2409	4102.0	1.0	0.0	738	2.0	0.538569	0.442681	0.233333	Shamil	

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR	first_name	ni
2410	4104.0	4.0	1.0	123	4.0	0.583490	0.444444	0.625000	David	
2411	4106.0	0.0	0.0	110	0.0	0.459459	0.380645	NaN	Danny	

2412 rows × 14 columns

In [12]:
```
#Rearranging the columns in df9
df9 = df9[['first_name', 'nickname', 'last_name','fighter_id','TSR', 'SSR', 'TKR', 'kno
```

In [13]:
```
df8
```

Out[13]:

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR	reach_cm	win
0	2.0	1.0	0.0	0	0.0	0.632653	0.526786	NaN	177.80	1
1	6.0	0.0	0.0	0	0.0	0.000000	0.000000	NaN	NaN	
2	7.0	4.0	4.0	1866	0.0	0.747646	0.613559	0.650000	172.72	1
3	9.0	1.0	0.0	166	0.0	0.591398	0.439394	0.000000	NaN	2
4	11.0	0.0	3.0	720	0.0	0.590551	0.470842	0.238095	190.50	1
...	...	...	...	...	...	...	...	...	...	...
2407	4099.0	3.0	0.0	804	0.0	0.613208	0.551136	0.571429	NaN	1
2408	4100.0	0.0	0.0	22	1.0	0.339806	0.336614	0.500000	180.34	
2409	4102.0	1.0	0.0	738	2.0	0.538569	0.442681	0.233333	193.04	2
2410	4104.0	4.0	1.0	123	4.0	0.583490	0.444444	0.625000	NaN	1
2411	4106.0	0.0	0.0	110	0.0	0.459459	0.380645	NaN	NaN	

2412 rows × 11 columns

In [14]:
```
#Filling missing values in reach_cm with the median
median_reach = df8['reach_cm'].median()

df8['reach_cm'] = df8['reach_cm'].fillna(median_reach)
```

In [15]:
```
#Filling null values with 0
df8['TSR'] = df8['TSR'].fillna(0)
df8['SSR'] = df8['SSR'].fillna(0)
df8['TKR'] = df8['TKR'].fillna(0)
```

In [16]:
```
df8
```

Out[16]:

	fighter_id	submiss_att	reversals	ctrl	knockdowns	TSR	SSR	TKR	reach_cm	win
0	2.0	1.0	0.0	0	0.0	0.632653	0.526786	0.000000	177.80	1
1	6.0	0.0	0.0	0	0.0	0.000000	0.000000	0.000000	182.88	
2	7.0	4.0	4.0	1866	0.0	0.747646	0.613559	0.650000	172.72	1
3	9.0	1.0	0.0	166	0.0	0.591398	0.439394	0.000000	182.88	2
4	11.0	0.0	3.0	720	0.0	0.590551	0.470842	0.238095	190.50	1
...	...	...	...	...	...	...	...	...	...	
2407	4099.0	3.0	0.0	804	0.0	0.613208	0.551136	0.571429	182.88	1
2408	4100.0	0.0	0.0	22	1.0	0.339806	0.336614	0.500000	180.34	
2409	4102.0	1.0	0.0	738	2.0	0.538569	0.442681	0.233333	193.04	2
2410	4104.0	4.0	1.0	123	4.0	0.583490	0.444444	0.625000	182.88	1
2411	4106.0	0.0	0.0	110	0.0	0.459459	0.380645	0.000000	182.88	

2412 rows × 11 columns

In [17]:
```
#Rearranging columns in df8
df8 = df8[['fighter_id','TSR', 'SSR', 'TKR', 'knockdowns','submiss_att','reversals', 'r
```

In [18]:
```
#Load fight dataset in df2
df2 = pd.read_csv("C:/Users/rohan/Downloads/UFC/UFC_FIGHT.csv")
```

In [19]:
```
df2
```

Out[19]:

	fight_id	event_id	referee	f_1	f_2	winner	num_rounds	title_fight	weight_class	genc
0	1	1	John McCarthy	1593.0	3295.0	1593.0	N	F	Open Weight	
1	2	1	John McCarthy	640.0	108.0	640.0	N	F	Open Weight	
2	3	1	John McCarthy	1060.0	2038.0	1060.0	N	F	Open Weight	
3	4	1	John McCarthy	2663.0	1952.0	2663.0	N	F	Open Weight	
4	5	1	John McCarthy	141.0	1962.0	141.0	N	F	Open Weight	
...	...	...	...	...	...	...	...	...	...	

	fight_id	event_id	referee	f_1	f_2	winner	num_rounds	title_fight	weight_class	genc
7213	7214	664	Herb Dean	1108.0	2320.0	2320.0	3	F	Featherweight	
7214	7215	664	Dan Miragliotta	3831.0	2974.0	3831.0	3	F	Welterweight	
7215	7216	664	Kerry Hatley	981.0	179.0	981.0	3	F	Women's Strawweight	
7216	7217	664	Mark Smith	1662.0	2464.0	1662.0	3	F	Featherweight	
7217	7218	664	Herb Dean	2976.0	2884.0	2884.0	5	F	Lightweight	

7218 rows × 15 columns

In [20]:
```
#Checking missing values
missing99 = df2.isna().sum()
missing99
```

Out[20]:
```
fight_id           0
event_id           0
referee           32
f_1               19
f_2               13
winner            15
num_rounds         0
title_fight        0
weight_class      13
gender             0
result             0
result_details    17
finish_round       0
finish_time        0
fight_url          0
dtype: int64
```

In [21]:
```
#Dropping unnecessary coluns
df2 = df2.drop(['num_rounds','title_fight', 'weight_class', 'gender','result_details',
        'finish_round', 'finish_time', 'fight_url','event_id' ], axis=1)
```

In [22]:
```
df2
```

Out[22]:

	fight_id	referee	f_1	f_2	winner
0	1	John McCarthy	1593.0	3295.0	1593.0
1	2	John McCarthy	640.0	108.0	640.0
2	3	John McCarthy	1060.0	2038.0	1060.0

	fight_id	referee	f_1	f_2	winner
3	4	John McCarthy	2663.0	1952.0	2663.0
4	5	John McCarthy	141.0	1962.0	141.0
...	...	...	...	...	...
7213	7214	Herb Dean	1108.0	2320.0	2320.0
7214	7215	Dan Miragliotta	3831.0	2974.0	3831.0
7215	7216	Kerry Hatley	981.0	179.0	981.0
7216	7217	Mark Smith	1662.0	2464.0	1662.0
7217	7218	Herb Dean	2976.0	2884.0	2884.0

7218 rows × 5 columns

In [23]:
```
#Merging fight df2 with stat df8 on f_1 using left join
df2 = pd.merge(df2, df8[['fighter_id','submiss_att','reversals','ctrl','knockdowns','TS
```

In [24]:
```
df2
```

Out[24]:

	fight_id	referee	f_1	f_2	winner	fighter_id	submiss_att	reversals	ctrl	knockdowns
0	1	John McCarthy	1593.0	3295.0	1593.0	1593.0	1.0	0.0	0.0	0.(
1	2	John McCarthy	640.0	108.0	640.0	640.0	3.0	0.0	0.0	1.(
2	3	John McCarthy	1060.0	2038.0	1060.0	1060.0	1.0	0.0	0.0	1.(
3	4	John McCarthy	2663.0	1952.0	2663.0	2663.0	3.0	1.0	0.0	0.(
4	5	John McCarthy	141.0	1962.0	141.0	141.0	0.0	0.0	0.0	0.(
...	...	...	...	...	...	...	...	...	...	..
7213	7214	Herb Dean	1108.0	2320.0	2320.0	1108.0	5.0	0.0	811.0	3.(
7214	7215	Dan Miragliotta	3831.0	2974.0	3831.0	3831.0	2.0	0.0	497.0	2.(
7215	7216	Kerry Hatley	981.0	179.0	981.0	981.0	2.0	1.0	641.0	2.(
7216	7217	Mark Smith	1662.0	2464.0	1662.0	1662.0	11.0	1.0	3179.0	1.(
7217	7218	Herb Dean	2976.0	2884.0	2884.0	2976.0	0.0	0.0	463.0	2.(

7218 rows × 16 columns

In [25]:
```
#Column names
df2.columns
```

Out[25]:
```
Index(['fight_id', 'referee', 'f_1', 'f_2', 'winner', 'fighter_id',
       'submiss_att', 'reversals', 'ctrl', 'knockdowns', 'TSR', 'SSR', 'TKR',
       'reach_cm', 'wins', 'loss'],
      dtype='object')
```

In [26]:
```
#Renaming the names of TSR as f_1
df2 = df2.rename(columns={'TSR': 'f1_TSR','SSR':'f1_SSR','TKR':'f1_TKR','knockdowns':'f
```

In [27]:
```
#Merging fight df2 with stat df8 on f_2 using left join
df2 = pd.merge(df2, df8[['fighter_id','submiss_att','reversals','ctrl','knockdowns','TS
```

In [28]:
```
df2
```

Out[28]:

	fight_id	referee	f_1	f_2	winner	fighter_id_x	f1_sub	f1_reversals	f1_ctrl	f1_knockdo
0	1	John McCarthy	1593.0	3295.0	1593.0	1593.0	1.0	0.0	0.0	
1	2	John McCarthy	640.0	108.0	640.0	640.0	3.0	0.0	0.0	
2	3	John McCarthy	1060.0	2038.0	1060.0	1060.0	1.0	0.0	0.0	
3	4	John McCarthy	2663.0	1952.0	2663.0	2663.0	3.0	1.0	0.0	
4	5	John McCarthy	141.0	1962.0	141.0	141.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	
7213	7214	Herb Dean	1108.0	2320.0	2320.0	1108.0	5.0	0.0	811.0	
7214	7215	Dan Miragliotta	3831.0	2974.0	3831.0	3831.0	2.0	0.0	497.0	
7215	7216	Kerry Hatley	981.0	179.0	981.0	981.0	2.0	1.0	641.0	
7216	7217	Mark Smith	1662.0	2464.0	1662.0	1662.0	11.0	1.0	3179.0	
7217	7218	Herb Dean	2976.0	2884.0	2884.0	2976.0	0.0	0.0	463.0	

7218 rows × 27 columns

In [29]:
```
#Renaming the names of TSR as f_2
df2 = df2.rename(columns={'TSR': 'f2_TSR','SSR':'f2_SSR','TKR':'f2_TKR','knockdowns':'f
```

In [30]:
```
df2
```

Out[30]:

	fight_id	referee	f_1	f_2	winner	fighter_id_x	f1_sub	f1_reversals	f1_ctrl	f1_knockdo
0	1	John McCarthy	1593.0	3295.0	1593.0	1593.0	1.0	0.0	0.0	
1	2	John McCarthy	640.0	108.0	640.0	640.0	3.0	0.0	0.0	
2	3	John McCarthy	1060.0	2038.0	1060.0	1060.0	1.0	0.0	0.0	
3	4	John McCarthy	2663.0	1952.0	2663.0	2663.0	3.0	1.0	0.0	
4	5	John McCarthy	141.0	1962.0	141.0	141.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...
7213	7214	Herb Dean	1108.0	2320.0	2320.0	1108.0	5.0	0.0	811.0	
7214	7215	Dan Miragliotta	3831.0	2974.0	3831.0	3831.0	2.0	0.0	497.0	
7215	7216	Kerry Hatley	981.0	179.0	981.0	981.0	2.0	1.0	641.0	
7216	7217	Mark Smith	1662.0	2464.0	1662.0	1662.0	11.0	1.0	3179.0	
7217	7218	Herb Dean	2976.0	2884.0	2884.0	2976.0	0.0	0.0	463.0	

7218 rows × 27 columns

In [31]:
```
df2.columns
```

Out[31]:
```
Index(['fight_id', 'referee', 'f_1', 'f_2', 'winner', 'fighter_id_x', 'f1_sub',
       'f1_reversals', 'f1_ctrl', 'f1_knockdowns', 'f1_TSR', 'f1_SSR',
       'f1_TKR', 'f1_reach', 'f1_wins', 'f1_loss', 'fighter_id_y', 'f2_sub',
       'f2_reversals', 'f2_ctrl', 'f2_knockdowns', 'f2_TSR', 'f2_SSR',
       'f2_TKR', 'f2_reach', 'f2_wins', 'f2_loss'],
      dtype='object')
```

In [32]:
```
#Dropping duplicate and unnecessary columns
df2 = df2.drop(['fighter_id_x','fighter_id_y', 'referee'], axis=1)
```

In [34]:
```
df2
```

Out[34]:

	fight_id	f_1	f_2	winner	f1_sub	f1_reversals	f1_ctrl	f1_knockdowns	f1_TSR	f1_SSR
0	1	1593.0	3295.0	1593.0	1.0	0.0	0.0	0.0	1.000000	1.000000
1	2	640.0	108.0	640.0	3.0	0.0	0.0	1.0	0.685185	0.650000
2	3	1060.0	2038.0	1060.0	1.0	0.0	0.0	1.0	0.744361	0.509091
3	4	2663.0	1952.0	2663.0	3.0	1.0	0.0	0.0	0.933333	0.666667

	fight_id	f_1	f_2	winner	f1_sub	f1_reversals	f1_ctrl	f1_knockdowns	f1_TSR	f1_SSR
4	5	141.0	1962.0	141.0	0.0	0.0	0.0	0.0	0.727273	0.642857
...	...	...	...	...	...	...	...	...	...	...
7213	7214	1108.0	2320.0	2320.0	5.0	0.0	811.0	3.0	0.402412	0.377880
7214	7215	3831.0	2974.0	3831.0	2.0	0.0	497.0	2.0	0.582796	0.546835
7215	7216	981.0	179.0	981.0	2.0	1.0	641.0	2.0	0.556261	0.493827
7216	7217	1662.0	2464.0	1662.0	11.0	1.0	3179.0	1.0	0.712142	0.589681
7217	7218	2976.0	2884.0	2884.0	0.0	0.0	463.0	2.0	0.532119	0.516550

7218 rows × 24 columns

In [35]:
```
df2.columns
```

Out[35]:
```
Index(['fight_id', 'f_1', 'f_2', 'winner', 'f1_sub', 'f1_reversals', 'f1_ctrl',
       'f1_knockdowns', 'f1_TSR', 'f1_SSR', 'f1_TKR', 'f1_reach', 'f1_wins',
       'f1_loss', 'f2_sub', 'f2_reversals', 'f2_ctrl', 'f2_knockdowns',
       'f2_TSR', 'f2_SSR', 'f2_TKR', 'f2_reach', 'f2_wins', 'f2_loss'],
      dtype='object')
```

In [36]:
```
#Checking missing values
missing = df2.isna().sum()
missing
```

Out[36]:
```
fight_id          0
f_1              19
f_2              13
winner           15
f1_sub           19
f1_reversals     19
f1_ctrl          19
f1_knockdowns    19
f1_TSR           19
f1_SSR           19
f1_TKR           19
f1_reach         19
f1_wins          19
f1_loss          19
f2_sub           13
f2_reversals     13
f2_ctrl          13
f2_knockdowns    13
f2_TSR           13
f2_SSR           13
f2_TKR           13
f2_reach         13
f2_wins          13
f2_loss          13
dtype: int64
```

In [37]:
```
#Drop rows with missing values
df2 = df2.dropna()
```

In [38]:
```
#Checking missing values
missing1 = df2.isna().sum()
missing1
```

Out[38]:
```
fight_id         0
f_1              0
f_2              0
winner           0
f1_sub           0
f1_reversals     0
f1_ctrl          0
f1_knockdowns    0
f1_TSR           0
f1_SSR           0
f1_TKR           0
f1_reach         0
f1_wins          0
f1_loss          0
f2_sub           0
f2_reversals     0
f2_ctrl          0
f2_knockdowns    0
f2_TSR           0
f2_SSR           0
f2_TKR           0
f2_reach         0
f2_wins          0
f2_loss          0
dtype: int64
```

In [39]:
```
#Coputing the difference between stats of both the fighters
df2['TSR'] = df2['f1_TSR'] - df2['f2_TSR']
df2['SSR'] = df2['f1_SSR'] - df2['f2_SSR']
df2['TKR'] = df2['f1_TKR'] - df2['f2_TKR']
df2['knockdowns'] = df2['f1_knockdowns'] - df2['f2_knockdowns']
df2['sub'] = df2['f1_sub'] - df2['f2_sub']
df2['reversals'] = df2['f1_reversals'] - df2['f2_reversals']
df2['reach'] = df2['f1_reach'] - df2['f2_reach']
df2['wins'] = df2['f1_wins'] - df2['f2_wins']
df2['loss'] = df2['f1_loss'] - df2['f2_loss']
```

```
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['TSR'] = df2['f1_TSR'] - df2['f2_TSR']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:2: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['SSR'] = df2['f1_SSR'] - df2['f2_SSR']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:3: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['TKR'] = df2['f1_TKR'] - df2['f2_TKR']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:4: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['knockdowns'] = df2['f1_knockdowns'] - df2['f2_knockdowns']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:5: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['sub'] = df2['f1_sub'] - df2['f2_sub']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:6: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['reversals'] = df2['f1_reversals'] - df2['f2_reversals']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:7: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['reach'] = df2['f1_reach'] - df2['f2_reach']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:8: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['wins'] = df2['f1_wins'] - df2['f2_wins']
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3057523355.py:9: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['loss'] = df2['f1_loss'] - df2['f2_loss']
```

In [40]:
```
df2['ctrl'] = df2['f1_ctrl'] - df2['f2_ctrl']
```

```
C:\Users\rohan\AppData\Local\Temp/ipykernel_11956/3911850071.py:1: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['ctrl'] = df2['f1_ctrl'] - df2['f2_ctrl']
```

In [41]:
```
df2
```

Out[41]:

	fight_id	f_1	f_2	winner	f1_sub	f1_reversals	f1_ctrl	f1_knockdowns	f1_TSR	f1_SSR
0	1	1593.0	3295.0	1593.0	1.0	0.0	0.0	0.0	1.000000	1.000000
1	2	640.0	108.0	640.0	3.0	0.0	0.0	1.0	0.685185	0.650000
2	3	1060.0	2038.0	1060.0	1.0	0.0	0.0	1.0	0.744361	0.509091
3	4	2663.0	1952.0	2663.0	3.0	1.0	0.0	0.0	0.933333	0.666667
4	5	141.0	1962.0	141.0	0.0	0.0	0.0	0.0	0.727273	0.642857
...	...	...	...	...	...	...	...	...	...	...
7213	7214	1108.0	2320.0	2320.0	5.0	0.0	811.0	3.0	0.402412	0.377880
7214	7215	3831.0	2974.0	3831.0	2.0	0.0	497.0	2.0	0.582796	0.546835
7215	7216	981.0	179.0	981.0	2.0	1.0	641.0	2.0	0.556261	0.493827
7216	7217	1662.0	2464.0	1662.0	11.0	1.0	3179.0	1.0	0.712142	0.589681
7217	7218	2976.0	2884.0	2884.0	0.0	0.0	463.0	2.0	0.532119	0.516550

7186 rows × 34 columns

In [42]:
```
#Dropping individual stats
df2 = df2.drop(['f1_sub', 'f1_reversals', 'f1_ctrl',
       'f1_knockdowns', 'f1_TSR', 'f1_SSR', 'f1_TKR', 'f1_reach', 'f1_wins',
       'f1_loss', 'f2_sub', 'f2_reversals', 'f2_ctrl', 'f2_knockdowns',
       'f2_TSR', 'f2_SSR', 'f2_TKR', 'f2_reach', 'f2_wins', 'f2_loss'], axis=1)
```

In [43]:
```
df2.columns
```

Out[43]:
```
Index(['fight_id', 'f_1', 'f_2', 'winner', 'TSR', 'SSR', 'TKR', 'knockdowns',
       'sub', 'reversals', 'reach', 'wins', 'loss', 'ctrl'],
      dtype='object')
```

In [44]:
```
#Rearranging columns
df2 = df2[['fight_id', 'f_1', 'f_2', 'TSR', 'SSR', 'TKR', 'knockdowns',
          'sub', 'reversals', 'reach', 'wins', 'loss', 'ctrl','winner']]
```

In [45]:
```
df2
```

Out[45]:

	fight_id	f_1	f_2	TSR	SSR	TKR	knockdowns	sub	reversals	reach	wins
0	1	1593.0	3295.0	0.800000	1.000000	0.500000	0.0	1.0	0.0	0.00	2.0
1	2	640.0	108.0	-0.314815	-0.350000	0.666667	1.0	3.0	0.0	0.00	20.0
2	3	1060.0	2038.0	-0.187012	-0.290909	1.000000	1.0	1.0	0.0	0.00	1.0
3	4	2663.0	1952.0	0.933333	0.666667	1.000000	0.0	3.0	1.0	0.00	1.0
4	5	141.0	1962.0	0.393939	0.309524	-1.000000	0.0	-1.0	0.0	0.00	1.0
...	...	...	...	...	...	...	...	...	...	...	...
7213	7214	1108.0	2320.0	-0.156728	-0.122433	0.427083	-3.0	3.0	0.0	7.62	1.0
7214	7215	3831.0	2974.0	0.002460	0.058018	-0.083333	2.0	-1.0	-3.0	25.40	1.0
7215	7216	981.0	179.0	0.034340	0.034106	0.043011	2.0	-6.0	0.0	7.62	-1.0
7216	7217	1662.0	2464.0	0.212413	0.134798	0.193793	-4.0	10.0	-1.0	-2.54	-1.0
7217	7218	2976.0	2884.0	-0.041389	0.013817	0.093182	1.0	-1.0	0.0	2.54	-11.0

7186 rows × 14 columns

In [46]:
```
#Assigning binary integers to winners
#fighter 1: 1
#fighter 2: 0
df2['winner'] = (df2['f_1'] == df2['winner']).astype(int)
```

In [47]:
```
df2
```

Out[47]:

	fight_id	f_1	f_2	TSR	SSR	TKR	knockdowns	sub	reversals	reach	wins
0	1	1593.0	3295.0	0.800000	1.000000	0.500000	0.0	1.0	0.0	0.00	2.0
1	2	640.0	108.0	-0.314815	-0.350000	0.666667	1.0	3.0	0.0	0.00	20.0
2	3	1060.0	2038.0	-0.187012	-0.290909	1.000000	1.0	1.0	0.0	0.00	1.0
3	4	2663.0	1952.0	0.933333	0.666667	1.000000	0.0	3.0	1.0	0.00	1.0
4	5	141.0	1962.0	0.393939	0.309524	-1.000000	0.0	-1.0	0.0	0.00	1.0
...	...	...	...	...	...	...	...	...	...	...	...
7213	7214	1108.0	2320.0	-0.156728	-0.122433	0.427083	-3.0	3.0	0.0	7.62	1.0
7214	7215	3831.0	2974.0	0.002460	0.058018	-0.083333	2.0	-1.0	-3.0	25.40	1.0
7215	7216	981.0	179.0	0.034340	0.034106	0.043011	2.0	-6.0	0.0	7.62	-1.0
7216	7217	1662.0	2464.0	0.212413	0.134798	0.193793	-4.0	10.0	-1.0	-2.54	-1.0
7217	7218	2976.0	2884.0	-0.041389	0.013817	0.093182	1.0	-1.0	0.0	2.54	-11.0

7186 rows × 14 columns

In [48]:
```
#Drop fighter ids for analysis
df2 = df2.drop(['fight_id','f_1','f_2'], axis=1)
```

In [49]:
```
#Importing numpy and matplotlib
import numpy as np
import matplotlib.pyplot as plt
```

In [50]:
```
#Target variable is the last column
X = df2.iloc[:, :-1].values
y = df2.iloc[:, -1].values
```

In [51]:
```
#Splitting training and testing data - 75-25
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_stat
```

In [52]:
```
print(X_train)
```

```
[[-1.23806963e-01 -1.00823291e-01 -2.03506098e-01 ...  3.00000000e+00
   7.00000000e+00  1.15600000e+03]
 [-1.38666542e-01 -6.62557967e-02 -4.44444444e-01 ... -7.00000000e+00
   4.00000000e+00 -1.27000000e+02]
 [ 1.42753174e-01  9.34347627e-02 -6.90134321e-02 ... -5.00000000e+00
  -1.00000000e+00  2.16200000e+03]
 ...
 [-2.03201371e-01 -9.29454266e-02 -2.25297015e-02 ...  0.00000000e+00
  -4.00000000e+00 -2.13000000e+03]
 [ 2.02816955e-01  1.51021776e-01  3.44155844e-01 ...  6.00000000e+00
  -3.00000000e+00  2.99000000e+02]
 [-6.49717306e-02 -1.66939511e-01  4.35897436e-02 ...  4.50000000e+01
   8.00000000e+00  4.92000000e+02]]
```

In [53]:
```
print(y_train)
```

```
[0 0 0 ... 1 1 1]
```

In [54]:
```
print(X_test)
```

```
[[ 9.15906076e-03  1.59303858e-02  1.66666667e-01 ...  2.00000000e+00
   1.00000000e+00  1.09100000e+03]
 [-6.64095453e-02 -6.19855106e-02  6.85943775e-02 ...  0.00000000e+00
   4.00000000e+00  8.06000000e+02]
 [-1.53094353e-03 -7.56049879e-03  1.49056604e-01 ... -5.00000000e+00
   5.00000000e+00 -1.93000000e+02]
 ...
 [-7.63130010e-02 -9.77456638e-02  3.33951763e-02 ...  0.00000000e+00
  -2.00000000e+00  4.38000000e+02]
 [ 4.87416614e-02 -8.56657609e-02 -7.54901961e-02 ...  1.00000000e+01
   6.00000000e+00  6.17400000e+03]
```

```
 [ 2.86402277e-01  2.33103844e-01  3.30882353e-01 ... -2.00000000e+01
  -1.20000000e+01  4.92000000e+02]]
```

In [55]:
```
print(y_test)
```

```
[1 1 0 ... 1 0 0]
```

In [56]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [57]:
```
print(X_train)
```

```
[[-0.96697389 -0.89921437 -0.78686892 ...  0.09555108  1.15077383
   0.32627458]
 [-1.07379629 -0.62635658 -1.58449354 ... -0.80309599  0.62250532
  -0.32430095]
 [ 0.94927133  0.63415748 -0.34163195 ... -0.62336657 -0.25794221
   0.83639069]
 ...
 [-1.53772372 -0.83703061 -0.1877479  ... -0.17404304 -0.78621072
  -1.33996952]
 [ 1.38105733  1.08871935  1.02616189 ...  0.3651452  -0.61012121
  -0.10828756]
 [-0.54401967 -1.42110135  0.03114003 ...  3.86986876  1.32686333
  -0.01042234]]
```

In [58]:
```
print(X_test)
```

```
[[-1.11088750e-02  2.23783144e-02  4.38585276e-01 ...  5.68637317e-03
   9.42368028e-02  2.93314786e-01]
 [-5.54355820e-01 -5.92649169e-01  1.13917688e-01 ... -1.74043040e-01
   6.22505316e-01  1.48798789e-01]
 [-8.79570859e-02 -1.63046487e-01  3.80287293e-01 ... -6.23366574e-01
   7.98594820e-01 -3.57767812e-01]
 ...
 [-6.25549698e-01 -8.74921182e-01 -2.60900768e-03 ... -1.74043040e-01
  -4.34031710e-01 -3.78043231e-02]
 [ 2.73442184e-01 -7.79568724e-01 -3.63073228e-01 ...  7.24604027e-01
   9.74684324e-01  2.87077027e+00]
 [ 1.98193477e+00  1.73663242e+00  9.82220091e-01 ... -1.97133718e+00
  -2.19492675e+00 -1.04223447e-02]]
```

In [59]:
```
#Running the model
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

Out[59]:
```
LogisticRegression(random_state=0)
```

In [60]:
```
#Comparing the results
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[1 1]
 [1 1]
 [0 0]
 ...
 [1 1]
 [1 0]
 [1 0]]
```

In [61]:

```
#Confusion matrix and Accuracy
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

Out[61]:

```
[[279 334]
 [195 989]]
0.7056204785754034
```

In [62]:

```
df2
```

Out[62]:

	TSR	SSR	TKR	knockdowns	sub	reversals	reach	wins	loss	ctrl	winner
0	0.800000	1.000000	0.500000	0.0	1.0	0.0	0.00	2.0	-1.0	0.0	1
1	-0.314815	-0.350000	0.666667	1.0	3.0	0.0	0.00	20.0	16.0	0.0	1
2	-0.187012	-0.290909	1.000000	1.0	1.0	0.0	0.00	1.0	-2.0	0.0	1
3	0.933333	0.666667	1.000000	0.0	3.0	1.0	0.00	1.0	-1.0	0.0	1
4	0.393939	0.309524	-1.000000	0.0	-1.0	0.0	0.00	1.0	4.0	0.0	1
...	...	...	...	...	...	...	...	...	...	...	...
7213	-0.156728	-0.122433	0.427083	-3.0	3.0	0.0	7.62	1.0	-1.0	-127.0	0
7214	0.002460	0.058018	-0.083333	2.0	-1.0	-3.0	25.40	1.0	-1.0	-115.0	1
7215	0.034340	0.034106	0.043011	2.0	-6.0	0.0	7.62	-1.0	-9.0	-1723.0	1
7216	0.212413	0.134798	0.193793	-4.0	10.0	-1.0	-2.54	-1.0	-5.0	1862.0	1
7217	-0.041389	0.013817	0.093182	1.0	-1.0	0.0	2.54	-11.0	1.0	-832.0	0

7186 rows × 11 columns

In [97]:

```
# Input from user
input_id = int(input("Enter fighter ID: "))

# Fetch the row
row = df9[df9['fighter_id'] == input_id]

# Check if the row exists
if not row.empty:
    # Convert the row to a list
    row_as_list = row.values.tolist()[0]
    # Store it in a variable
    fighter_data = row_as_list
    print(f"Data for fighter {input_id}: {fighter_data}")
```

```
else:
    print(f"No data found for fighter ID {input_id}")


new_list = row_as_list[4:]
info_list = row_as_list[0:3]
f_name, n_name, l_name = row_as_list[0], row_as_list[1], row_as_list[2]
print(new_list)
```

```
Enter fighter ID: 1768
Data for fighter 1768: ['Conor', 'The Notorious', 'McGregor', 1768.0, 0.564048124557678
7, 0.49750830564784054, 0.5555555555555556, 13.0, 1.0, 1.0, 187.96, 22, 6, 833]
[0.5640481245576787, 0.49750830564784054, 0.5555555555555556, 13.0, 1.0, 1.0, 187.96, 2
2, 6, 833]
```

In [98]:

```
input_id1 = int(input("Enter fighter ID: "))

# Fetch the row
row1 = df9[df9['fighter_id'] == input_id1]

# Check if the row exists
if not row.empty:
    # Convert the row to a list
    row_as_list1 = row1.values.tolist()[0]
    # Store it in a variable
    fighter_data1 = row_as_list1
    print(f"Data for fighter {input_id1}: {fighter_data1}")
else:
    print(f"No data found for fighter ID {input_id1}")


new_list1 = row_as_list1[4:]
info_list1 = row_as_list1[0:3]
f_name1, n_name1, l_name1 = row_as_list1[0], row_as_list1[1], row_as_list1[2]
print(new_list1)
```

```
Enter fighter ID: 227
Data for fighter 227: ['Alexander', 'The Great', 'Volkanovski', 227.0, 0.601793424111590
9, 0.5699168556311414, 0.379746835443038, 6.0, 3.0, 3.0, 180.34, 26, 2, 3295]
[0.6017934241115909, 0.5699168556311414, 0.379746835443038, 6.0, 3.0, 3.0, 180.34, 26,
2, 3295]
```

In [99]:

```
difference = [a - b for a, b in zip(new_list, new_list1)]
print(difference)
```

```
[-0.03774529955391215, -0.07240854998330082, 0.17580872011251758, 7.0, -2.0, -2.0, 7.620
0000000000045, -4, 4, -2462]
```

In [100…

```
result = classifier.predict(sc.transform([difference]))
```

In [101…

```
if result == 1:
    print(f_name + " " + n_name +" "+ l_name, "wins")
else:
    print(f_name1 +" "+ n_name1 +" "+ l_name1, "wins")
```

```
Alexander The Great Volkanovski wins
```

In [ ]: