UFC Data: EDA

Вступление

UFC, или The Ultimate Fighting Championship, - это американский промоушен по смешанным единоборствам (MMA) с головным офисом в Лас Вегасе, штат Невада.

UFC произвел революцию в бойцовской индустрии и представляет собой глобальный спортивный бренд премиального класса, медиа компанию и крупнейший провайдер Pay-Per-View* в мире. На 2023 год UFC являлся крупнейшим MMA промоушеном в мире.
В рамках данного проекта по исследовательской аналитике данных мы проанализируем показатели бойцов и попытаемся установить, существует ли корреляция между антропометрическими данными и навыками в борьбе и стойке и количеством побед.

*Pay-per-view (PPV; букв. с англ. — «плата за просмотр») — вид платного телевидения или услуга веб-трансляции, которая позволяет зрителям платить за просмотр отдельных шоу через частные телеканалы.

Источник данных

Датасет доступен на сайте Kaggle по данной ссылке

Описание данных

name: имя и фамилия бойца UFC

nickname: прозвище бойца UFC

wins: количество побед за всю профессиональную карьеру

losses: количество поражений за всю профессиональую карьеру

draws: количество ничьих за всю профессиональую карьеру

height_cm: рост бойца в сантиметрах

weight_in_kg: вес бойца в килограммах

reach_in_cm: размах рук в сантиметрах

stance: тип стойки

date_of_birth: дата рождения бойца

significant_strikes_landed_per_minute: количество значимых ударов, наносимых бойцом в минуту

significant_striking_accuracy: коэффициент точности значимых ударов

significant_strikes_absorbed_per_minute: количество значимых ударов, пропускаемых бойцом в минуту

significant_strike_defence: коэффициент защиты от значимых ударов

average_takedowns_landed_per_15_minutes: количество тейкдаунов за 15 минут

takedown_accuracy: коэффициент точности тейкдаунов (доля успешно проведенных тейкдаунов)

takedown_defense: коэффициент защиты от тейкдаунов

average_submissions_attempted_per_15_minutes: количество попыток сабмишена, предпринимаемых бойцом за 15 минут

Предварительные шаги

Прежде всего, изменим наименования колонок, чьи названия чересчур длинные, т.к. это усложняет восприятие, а также удалим те колонки, что нам не пригодятся в анализе.

▼ Код

```
df.rename(columns={subset_col[0]: 'sig_strikes_minute', subset_col[1]: 'sig_s
trike_accuracy', subset_col[2]: 'sig_strikes_absorb_min', subset_col[3]: 'sig
_strike_defence', subset_col[4]: 'avg_takedowns_per_15min', subset_col[5]: 'a
vg_sub_per_15min'}, inplace=True)
```

▼ Код

```
df.drop(columns=['nickname', 'date_of_birth'], axis=1, inplace=True)
```

Мы собираемся анализировать физические данные бойцов, поэтому необходимо удалить строки со значениями NaN в колонках **height_cm, weight_in_kg, reach_in_cm**.
В колонке
stance эти значения мы заменим на 'free' (свободная стойка).

▼ Код

```
subset_cols = ['stance']
[df[col].fillna('Free', inplace=True) for col in subset_cols]
```

▼ Код

```
df.dropna(subset=['height_cm','weight_in_kg','reach_in_cm'], inplace=True)
df.isna().sum()
```

Конструирование признаков

Поскольку в датасете нет информации, касательно весовой категории бойцов, мы создадим этот признак вручную. Это будет не абсолютно точный признак, но такая апроксимация позволит нам осуществить анализ.

▼ Код

```
#создаем словарь, по которому будем далее присваивать значение весовой катего
рии
weight_class = pd.DataFrame({'from': [0, 58, 62, 67, 71, 85, 94],
'to': [57.9, 61.9, 66.9, 70.9, 84.9, 93.9, 120.9],
'value': ['Flyweight', 'Bantamweight', 'Featherweight', 'Lightweight', 'Middl
eweight', 'Light Heavyweight', 'Heavyweight']})

#присваиваем значения
def assign_weight_class(x):
    return weight_class.loc[(x >= weight_class['from']) & (x <= weight_class
['to']), 'value'].squeeze()

df['weight_class'] = df['weight_in_kg'].apply(assign_weight_class)
```

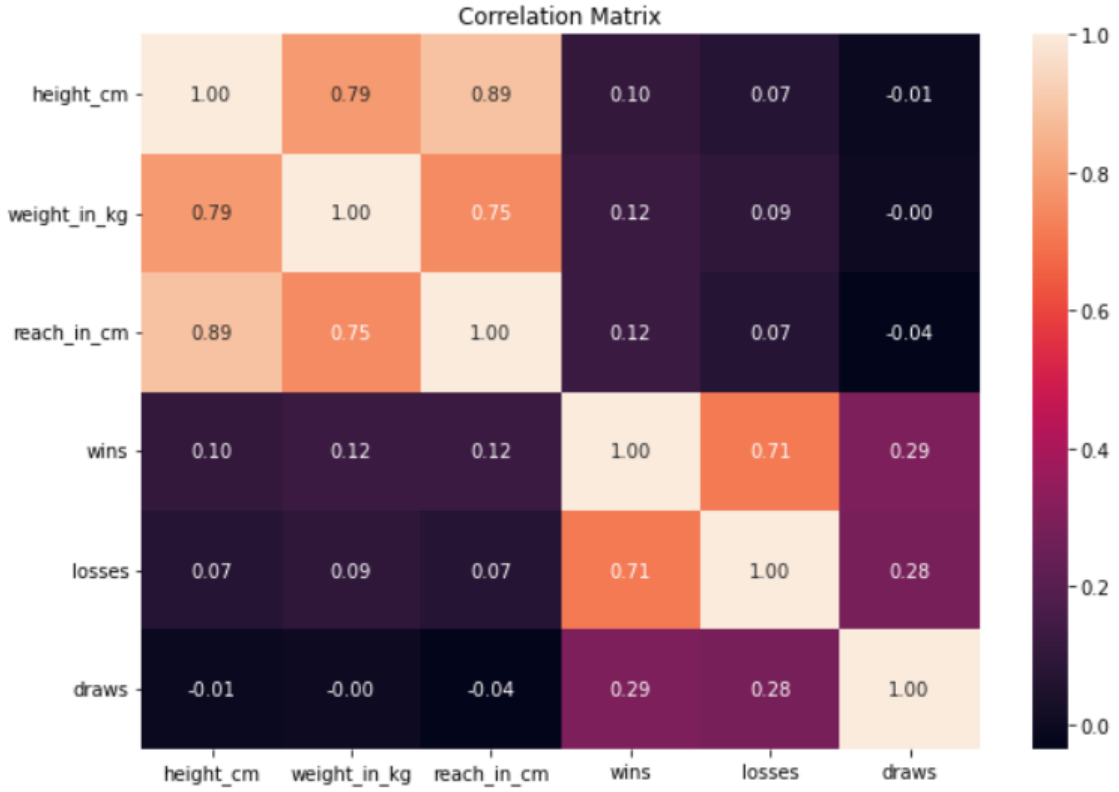
Анализ корреляции

Создадим 2 группы признаков и оценим корреляцию между ними:

1. Физические данные (антропометрия): рост, вес, размах рук.
2. Рекорд: победы, поражения, ничьи.

▼ Код

```
physical_features = ['height_cm', 'weight_in_kg', 'reach_in_cm']
record = ['wins', 'losses', 'draws']

corr_matrix = df[physical_features + record].corr()

plt.figure(figsize=(10,7))
sns.heatmap(data = corr_matrix,
            annot=True,
            cmap="rocket",
            fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



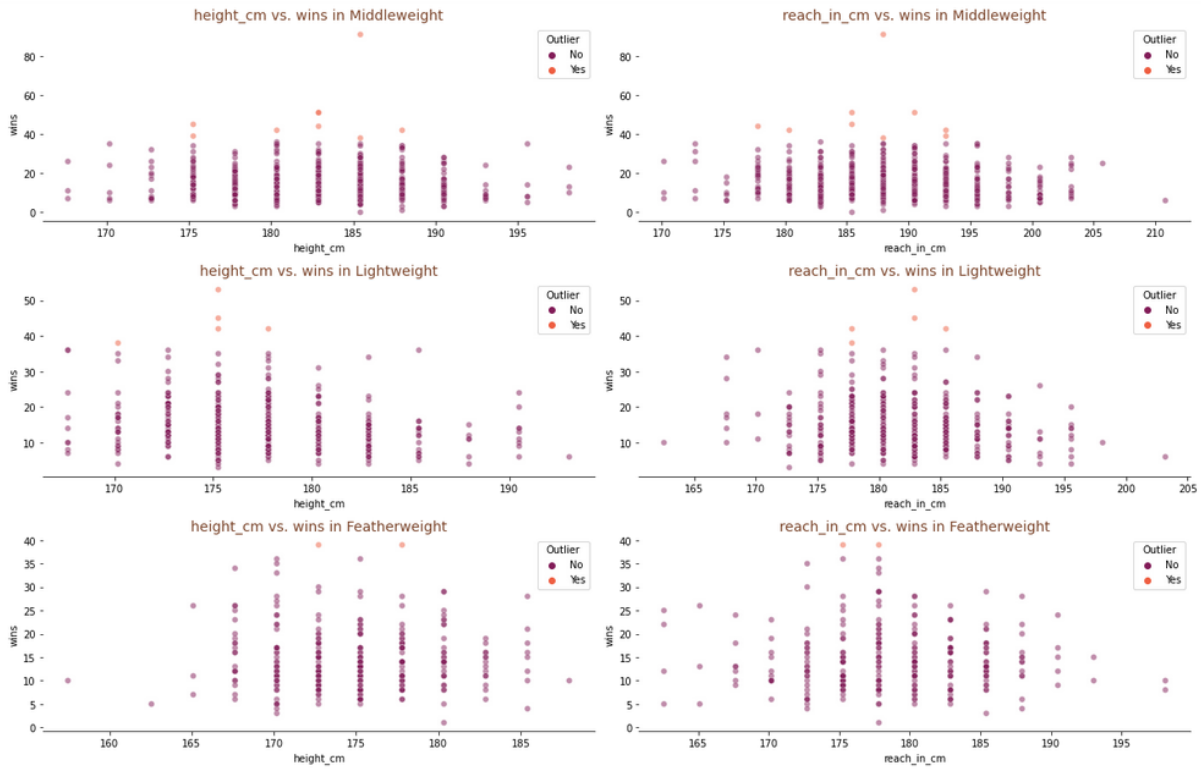
Корреляция между физическими данными (рост, вес, размах рук) и профессиональным рекордом выглядит довольно слабой. Интересно, что сильная позитивная корреляция наблюдается между количеством побед и поражений. Непобежденный боец - большая редкость. Среди всех бойцов UFC лишь 70 имеют статус "непобежденный".

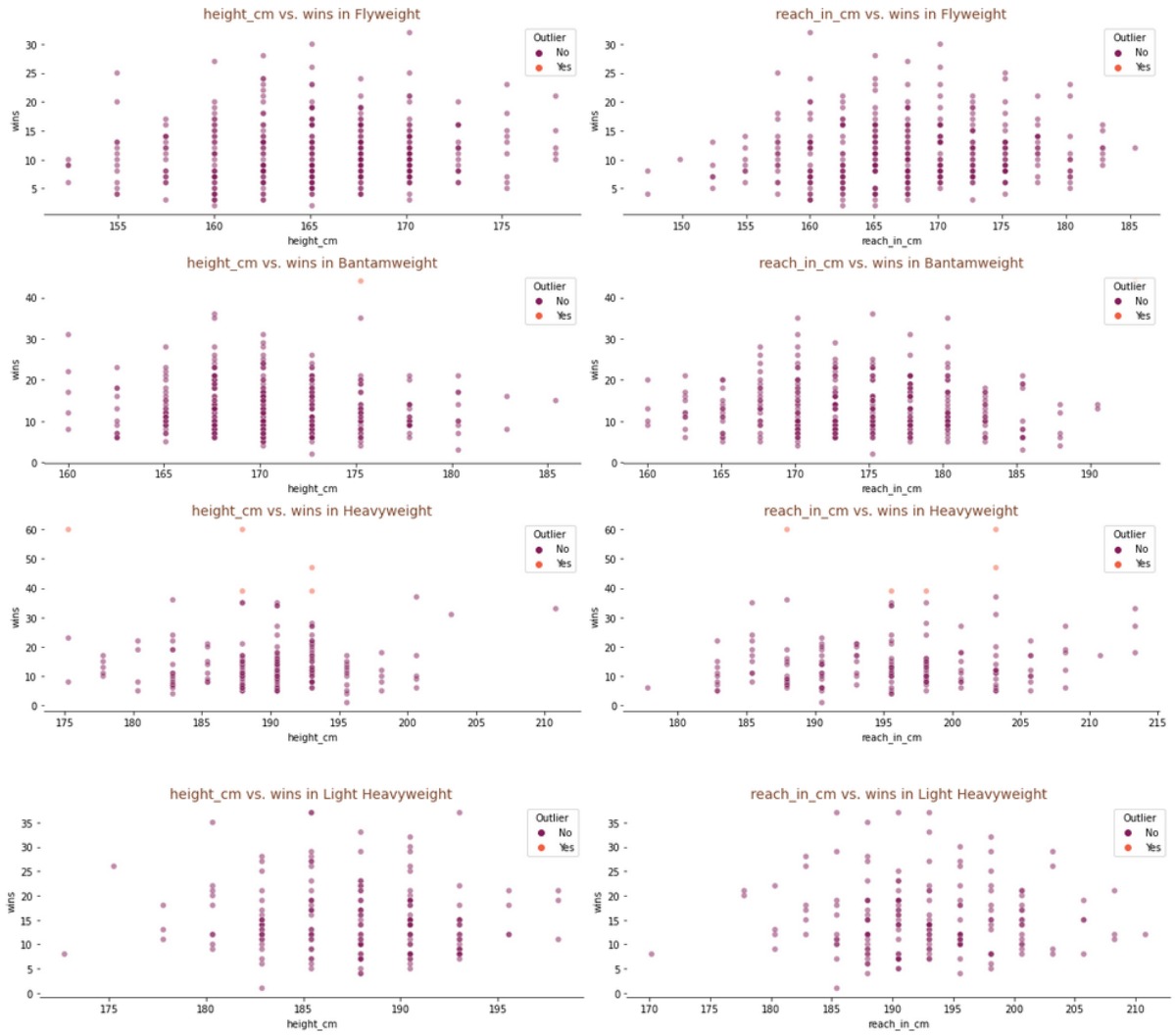
Анализ физических данных и количества побед в разрезе весовой категории

▼ Код

```
fig = plt.figure(figsize=(25, 25))
gs = GridSpec(ncols=3, nrows=7, figure=fig)

x=0
c=0
r=0
features = ['height_cm', 'reach_in_cm']   #не включаем вес, т.к. будем исполь
зовать весовую категорию
wc = list(df['weight_class'].unique())

for weight in wc:
    for i, feature in enumerate(features):
        plt.subplot(gs[c,r])
        is_outlier = (df['wins'] >= df['wins'].mean() + df['wins'].std()*3)
        dt = df[df['weight_class'] == wc[x]]
        ax = sns.scatterplot(data = dt, x = df[df['weight_class'] == wc[x]][f
eature], y = df[df['weight_class'] == wc[x]]['wins'], hue = is_outlier, palet
```

```
te = 'rocket', alpha=0.5)
        handles, labels  =  ax.get_legend_handles_labels()
        ax.legend(handles, ['No', 'Yes'], loc='upper right', title = 'Outlie
r')
        plt.title(f'{feature} vs. wins in {weight}', color='#7A442A', fontsiz
e=14)
        r+=1
    r=0
    x+=1
    c+=1

sns.despine(bottom=False, left=True)
plt.tight_layout()
plt.show()
```





Весовая категория и количество побед

▼ Код

```
a = df.groupby('weight_class')['wins'].mean().round(2).reset_index()

fig = plt.figure(figsize=(10,7))

ax = sns.barplot(data = a, x = 'weight_class', y = 'wins', palette='rocket',
order=a.sort_values('wins').weight_class)

plt.xticks(rotation=45)

sns.despine(left=True, right=False)

ax.set(ylim=(8, 20))

for i in ax.containers:
    ax.bar_label(i,)

plt.title('Avg wins by weight_class', color = '#7A442A')
```

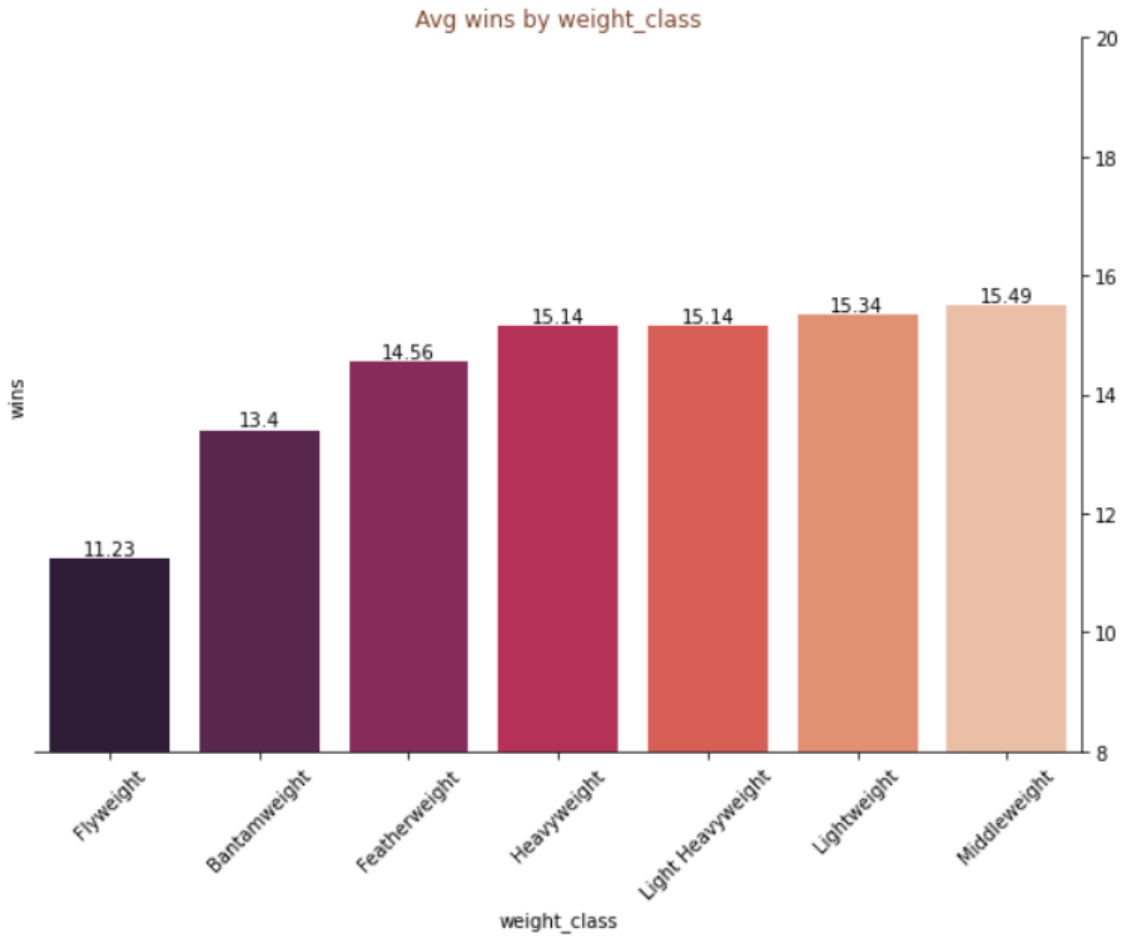


График Avg wins by weight_class показывает, что среднее количество побед выше всего в среднем весе и ниже всего в наилегчайшем весе. Давайте проанализируем эту разницу подробнее

Т.к. дисперсия выборок и их размеры разные, мы будем использовать t-тест Уэлча, чтобы определить, есть ли статистически значимая разница между средним количеством побед в средней и наилегчайшей весовой категории.

Нулевая гипотеза H0: среднее количество побед в среднем весе = среднее количество побед в наилегчайшем весе

Альтернативная гипотеза H1: среднее количество побед в среднем весе > среднее количество побед в наилегчайшем весе

▼ Код

```
mw = df.query('weight_class == "Middleweight"')['wins']
fw = df.query('weight_class == "Flyweight"')['wins']

alpha = 0.05
t_crit, p_value = ttest_ind(mw, fw, equal_var=False)

if alpha <  p_value:
    print('среднее количество побед в среднем весе выше \nT-statictic: {:.2f}
\np-value: {:.2f}'.format(t_crit, p_value))
else:
    print('Статистически значимая разница не обнаружена \nT-statictic: {:.2
```

```
f}, p-value: {:.2f}'.format(t_crit, p_value))
```

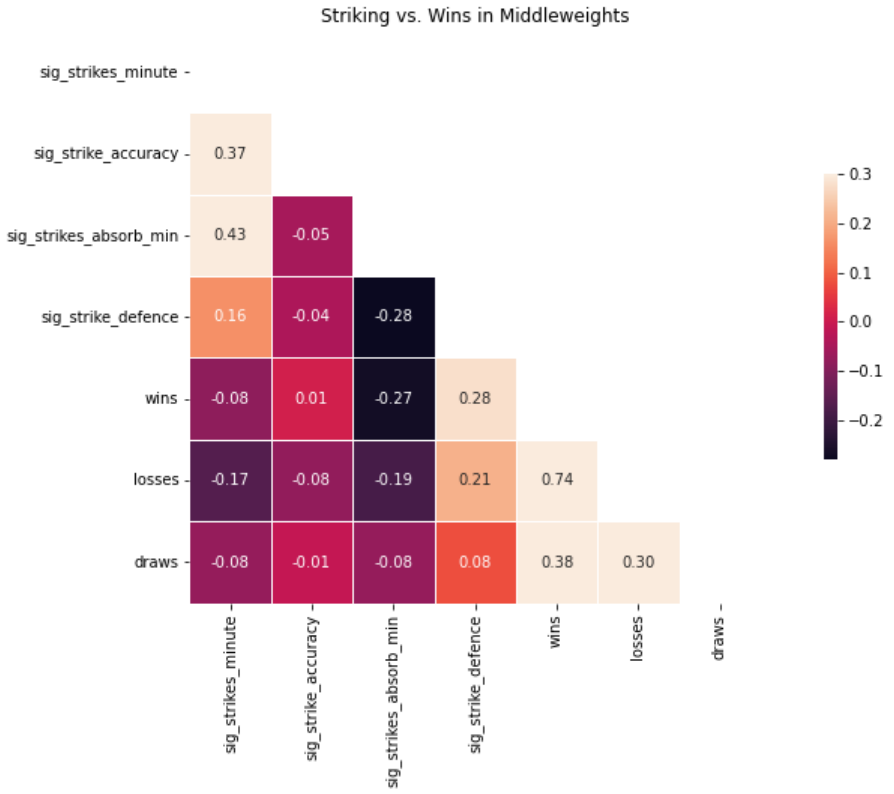
Статистически значимой разницы не обнаружено, оснований отвергнуть нулевую гипотезу при уровне значимости 0.05 нет.

Весовая категория + Ударная техника/Грэпплинг vs Количество побед

Сильная положительная корреляция (coef > 0.3) обнаружена между защитой от значимых ударов и количеством побед в средней весовой категории.

▼ Код

```
mw = df.query('weight_class=="Middleweight"')[striking_stats + record]

corr_matrix = mw[striking_stats + record].corr()

plt.figure(figsize = (15,7))

mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
sns.heatmap(corr_matrix, annot=True, fmt='.2f', mask=mask, cmap="rocket", vma
x=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.title('Striking vs. Wins in Middleweights')

plt.show()
```



▼ Код

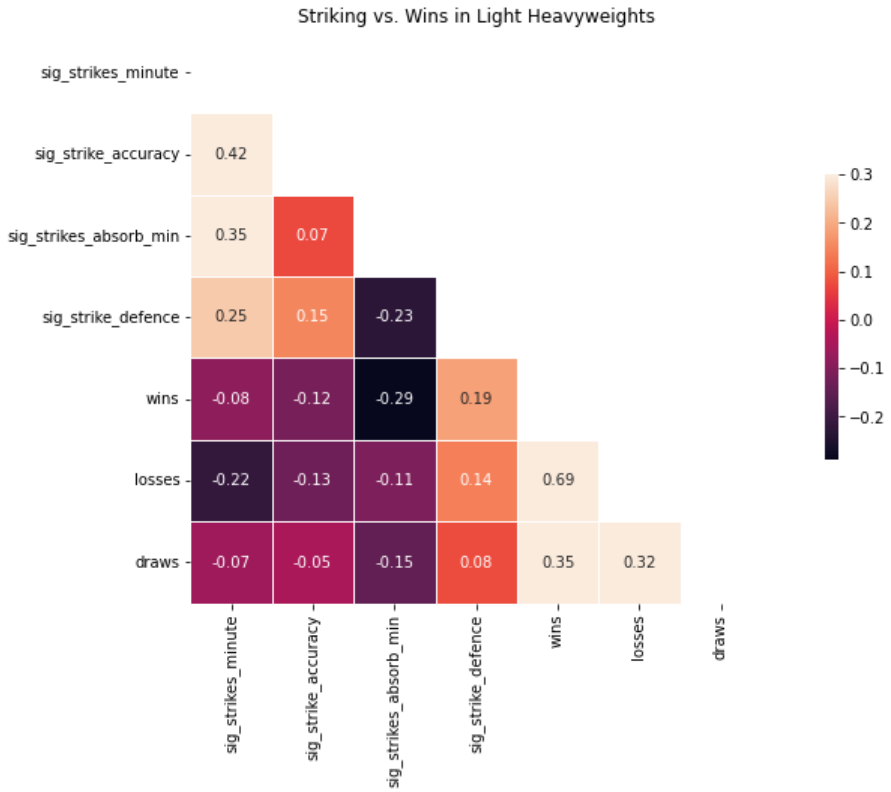
```
striking = list(mw[striking_stats].columns)
rec = list(mw[record].columns)
striking_len = len(striking)
rec_len = len(rec)
l = []
coefs = []
striking_record_corr = {}
x=0

for i in range(striking_len):
    for j in range(rec_len):
        striking_rec = striking[i]+' vs '+rec[j]
        l.append(striking_rec)
        spearman_coef = spearmanr(mw[striking[i]], mw[record[j]]).correlation
        coefs.append(spearman_coef)

for i in l:
    striking_record_corr[l[x]] = coefs[x]
    x+=1

t = pd.DataFrame(striking_record_corr.items(), columns=['stats', 'corr_coe
f'])
pd.set_option('display.max_colwidth', None)
t.sort_values(by='corr_coef', ascending=False)
```

	stats	corr_coef
9	sig_strike_defence vs wins	0.323188
10	sig_strike_defence vs losses	0.189869
11	sig_strike_defence vs draws	0.057765
0	sig_strikes_minute vs wins	-0.022034
5	sig_strike_accuracy vs draws	-0.042504
3	sig_strike_accuracy vs wins	-0.049990
2	sig_strikes_minute vs draws	-0.056891
8	sig_strikes_absorb_min vs draws	-0.083843
1	sig_strikes_minute vs losses	-0.136692
4	sig_strike_accuracy vs losses	-0.156078
7	sig_strikes_absorb_min vs losses	-0.164290
6	sig_strikes_absorb_min vs wins	-0.262085

Сильная отрицательная корреляция (coef>|0.3|) обнаружена между количеством пропущенных значимых ударов и побед в полутяжёлом весе.

▼ Код

```
lhw = df.query('weight_class=="Light Heavyweight"')[striking_stats + record]

corr_matrix = lhw[striking_stats + record].corr()

plt.figure(figsize = (15,7))

mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
sns.heatmap(corr_matrix, annot=True, fmt='.2f', mask=mask, cmap="rocket", vma
x=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.title('Striking vs. Wins in Light Heavyweights')

plt.show()
```



▼ Код

```
striking = list(lhw[striking_stats].columns)
rec = list(lhw[record].columns)
striking_len = len(striking)
rec_len = len(rec)
l = []
coefs = []
striking_record_corr = {}
x=0

for i in range(striking_len):
```

```
    for j in range(rec_len):
        striking_rec = striking[i]+' vs '+rec[j]
        l.append(striking_rec)
        spearman_coef = spearmanr(lhw[striking[i]], lhw[record[j]]).correlati
on
        coefs.append(spearman_coef)

for i in l:
    striking_record_corr[l[x]] = coefs[x]
    x+=1

t = pd.DataFrame(striking_record_corr.items(), columns=['stats', 'corr_coe
f'])
pd.set_option('display.max_colwidth', None)
t.sort_values(by='corr_coef', ascending=False)
```

	stats	corr_coef
9	sig_strike_defence vs wins	0.214402
10	sig_strike_defence vs losses	0.179109
11	sig_strike_defence vs draws	0.132196
2	sig_strikes_minute vs draws	0.001499
5	sig_strike_accuracy vs draws	-0.022034
0	sig_strikes_minute vs wins	-0.054499
7	sig_strikes_absorb_min vs losses	-0.130587
3	sig_strike_accuracy vs wins	-0.139284
8	sig_strikes_absorb_min vs draws	-0.166686
4	sig_strike_accuracy vs losses	-0.176550
1	sig_strikes_minute vs losses	-0.192814
6	sig_strikes_absorb_min vs wins	-0.315041

В среднем, средневесы демонстрируют борцовские навыки лучше, чем у других весовых категорий, за исключением лишь защиты от тейкдаунов, где лучший показатель у полутяжеловесов.

▼ Код

```
df.groupby('weight_class')[grappling_stats].mean().round(2)
```

weight_class	avg_takedowns_per_15min	takedown_accuracy	takedown_defense	avg_sub_per_15min
Bantamweight	1.51	32.13	54.68	0.53
Featherweight	1.57	32.51	53.52	0.62
Flyweight	1.53	35.31	51.11	0.60
Heavyweight	1.49	33.54	54.01	0.49
Light Heavyweight	1.50	31.64	55.02	0.48
Lightweight	1.50	32.61	52.94	0.64
Middleweight	1.64	37.51	53.77	0.70

Заключения

Наши наблюдения по итогам анализа датасета:

- Конкретный тип стойки не коррелирует с большим количеством побед.
- Сильная позитивная корреляция обнаружена между ударной техникой и количеством побед в среднем весе.
- Сильная отрицательная корреляция обнаружена между количеством пропущенных значимых ударов и количеством побед в полутяжёлом весе.
- В среднем, средневесы демонстрируют лучший уровень борьбы, единственное исключение - защита от тейкдаунов, где первенство перехватывают полутяжелом весе.
- Позитивная корреляция обнаружена между средним количеством попыток сабмишена и количеством побед в полулёгком весе.
- Сильная позитивная корреляция обнаружена между средним количеством попыток сабмишена и количеством поражений в полутяжелом весе.

Ссылки

Ссылка на полный блокнот Jupyter

Контакты

lianazaripovar@gmail.com

tg: @zaripova_liana