MMAPredict

An Analytical Approach to Predicting UFC Fight Outcomes

By Simon Babich, Sunny Rahman, & Chris McLaughlin

Overview

- We began this project with the goal of developing a predictive model for UFC fights that could accurately predict the outcome of a fight with greater than 66.31% accuracy[1]
    - [1]The winning rate for favored fighters, according to sports betting site Pinnacle
- We also sought to create a risk analysis score by comparing the predictions of this model to lines set by sportsbooks in order to evaluate whether or not it is "safe" to bet on that fight
- We also wanted to create a prototype of an App interface where users could explore this data for upcoming UFC matchups to help guide their betting decisions, or just to understand the matchup a bit better
- We realized that MMA is a very volatile sport where entire fights can change instantly off of just a single move or punch, making it hard to predict and acknowledging that it would be impossible to accurately predict the winner every single time, so we decided that surpassing that 66.31% benchmark by even the thinnest of margins would make our project a success.

A Quick Refresher on MMA

- MMA is a combat sport where 2 fighters in the same weight class compete against each other in a cage with *limited* rules over three 5-Minute rounds (or 5 rounds for main events/championship bouts)
- A fighter can win either by:
  - Completely knocking out their opponent (KO),
  - The fight being stopped by the referee for their opponents safety (TKO)
  - Or by grappling their opponent until they tap out (Submission)
- If none of these things occur by the end of the fight, the outcome is decided by three judges who score the bout based on who they feel won each round.
  - If all three judges pick the same fighter, this is known as a Unanimous Decision (U-Dec),
  - If ⅔ judges pick the same fighter, this is known as a split decision (S-Dec.)
  - Rarely, 2 judges may score the bout as a draw and the third picks a fighter, that fighter wins in a Majority Draw (M-Dec)
- There are also outcomes such as Disqualification (DQ) or No Contest (NC), which can happen during the fight if a fighter injures their opponent on an illegal move or retroactively if one of the fighters is found to have been cheating somehow (NOTE: We omitted fights with these outcomes from our dataset to avoid skewing it)

Limitations and Re-focusing our vision

- Although we originally planned to create an AI-based ML model that would allow you to select any 2 fighters and instantly generate odds on the outcome probability as well as a betting risk assessment score, we realized that this scope was beyond our capabilities
- We also planned on characterizing fighters based on their fighting style archetype (i.e. Boxer, Wrestler, etc.), however limitations in the data set made this difficult to meaningfully discern different fighting styles, so we scrapped that aspect
- We eventually pivoted to creating a statistic based formula tailored to each weight class that would simply predict an expected winner and loser of a matchup

Our Data

- We started with a CSV file containing data on over 6000 UFC fights
- This data was sourced from a github repository that scraped MMA website Sherdog.com and UFC.com to retrieve fight data
- Using SQL and Excel, we eventually narrowed this down to 2745 Fights, with 59 points of data per fight, resulting in a total of 161,955 data points to work with
- This refined dataset removed all entries prior to 11/7/14, which was when betting data started being tracked, as well removing all fights that were missing data points
- We also removed Women's division and Catchweight fights from the dataset to avoid skewing our results due to a lack of significant data

Our Approach

- First, we needed to narrow down our dataset from all 59 columns to just the data points that impacted the outcome of the fight
- We eventually settled on 11 total, which are displayed in the right column

1. **Reach (In.):** The fighter's wingspan in inches
2. **Age:** The fighter's age during the fight
3. **Sig. Strikes Landed/Min:** The average number of significant strikes the fighter landed per minute
4. **Sig. Strikes Acc. %:** The percentage of significant strikes the fighter successfully landed
5. **Sig. Strikes Ate/Min:** The average number of significant strikes the fighter absorbed per minute
6. **Sig. Strikes Block %:** The percentage of the opponent's significant strikes that the fighter blocked
7. **Takedowns/15 Min.:** The amount of successful takedowns the fighter had per 15 minutes of fighting
8. **Takedown %:** The percentage of takedown attempts that the fighter successfully completed
9. **Takedown Def. %:** The percentage of the opponent's takedown attempts that the fighter successfully defended
10. **Sub. Attempts/15 min.:** The amount of submission attempts the fighter made per 15 minutes of fighting
11. **Is Favored:** A boolean representing whether the fighter was favored by the betting lines, where 1=Favored and 0=Underdog

Our Formula

- We first created a differential score of 11 different stats for each fight by subtracting the Loser's value from the Winner's value
- We then took the Z-Score of these differentials, converted them to absolute values, and took the mean of these Z-Scores
- We used the values generated by this formula as weights for each of these 11 statistics, then multiplied each fighter's stats by the corresponding weight, and added them together to generate the fighter score
- Then we compared each fighter's score in a bout to generate the "Expected Winner", and calculated our expected winning percentage by dividing the total number of fights by the amount of times the actual winner was also the expected winner

Example of our code

- This is an example of the code we used to calculate fighter scores and determine expected wins based on those scores

Overall

```
for stat in betsColumns:
    diff_column_name = f'{stat} Diff.'
    allMensWeights[diff_column_name] = allMensWeights[f'Winner {stat}'] - allMensWeights[f'Loser {stat}']

allMensWeights.dropna(subset=[f'{stat} Diff.' for stat in betsColumns], inplace=True)

from scipy.stats import zscore

allMensWeights_standardized = pd.DataFrame()

for stat in betsColumns:
    diff_col = f'{stat} Diff.'
    allMensWeights_standardized[diff_col] = zscore(allMensWeights[diff_col],nan_policy='omit')

impact_scores = allMensWeights_standardized.abs().mean().sort_values(ascending=False)
impact_scores
```

```
Is Favored Diff.                0.955809
Age Diff.                       0.798192
Reach (In.) Diff.               0.791910
Sig. Strikes Acc. % Diff.       0.785549
Sig. Strikes Block % Diff.      0.783655
Sig. Strikes Landed/Min Diff.   0.774803
Takedown Def. % Diff.           0.767091
Takedown % Diff.                0.757699
Sig. Strikes Ate/Min Diff.      0.754421
Takedowns/15 Min. Diff.         0.736593
Sub. Attempts/15 min. Diff.     0.659832
dtype: float64
```

```
betScoreF1 = ['Winner Is Favored','Winner Age','Winner Reach (In.)','Winner Sig. Strikes Acc. %','Winner Sig. Str
betScoreF2 = ['Loser Is Favored','Loser Age','Loser Reach (In.)','Loser Sig. Strikes Acc. %','Loser Sig. Strikes
weights_f1 = [0.955809,-0.10555,0.009,0.785549,0.783655,0.774803,0.767091,0.757699,-0.754421,0.736593,0.659832]
allMensWeights['Winner score'] = (allMensWeights[betScoreF1] * weights_f1).sum(axis=1)
allMensWeights['Loser score'] = (allMensWeights[betScoreF2] * weights_f1).sum(axis=1)
count_fighter2_wins = (allMensWeights['Loser score'] > allMensWeights['Winner score']).sum()
count_fighter1_wins = (allMensWeights['Winner score'] > allMensWeights['Loser score']).sum()
expectedWs=count_fighter1_wins/(count_fighter2_wins+count_fighter1_wins)
print('Actual Winner Expected Ws: ',count_fighter1_wins)
print('Actual Loser Expected Ws: ',count_fighter2_wins)
print('Accuracy of Expected Ws: ',expectedWs)
```

```
Actual Winner Expected Ws:  1819
Actual Loser Expected Ws:  926
Accuracy of Expected Ws:  0.6626593806921676
```

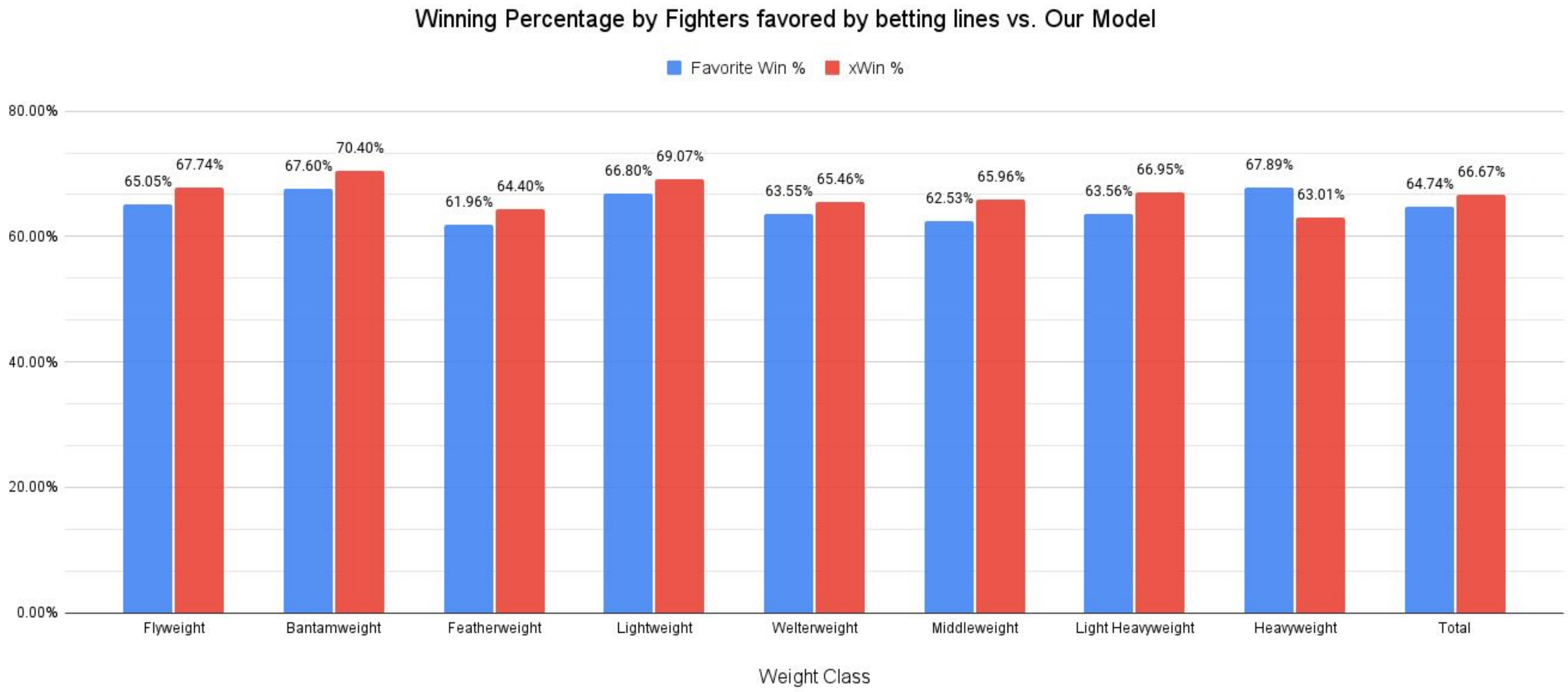
Our Conclusions

- When applied to the historical fight data upon which it was based, our model correctly predicted the winner a total of 1,830 times out of 2745 Fights (66.67%)
- Conversely, the fighter favored by betting lines won 1,777 times out of 2745 Fights (64.74%)
- This means our model outperformed betting lines by 1.93% across all weight classes
- Our model outperformed betting lines by 1.91% to 3.43% in every weight class, with the exception of heavyweight, where it underperformed betting lines by 4.88%

Breakdown by Weight Class

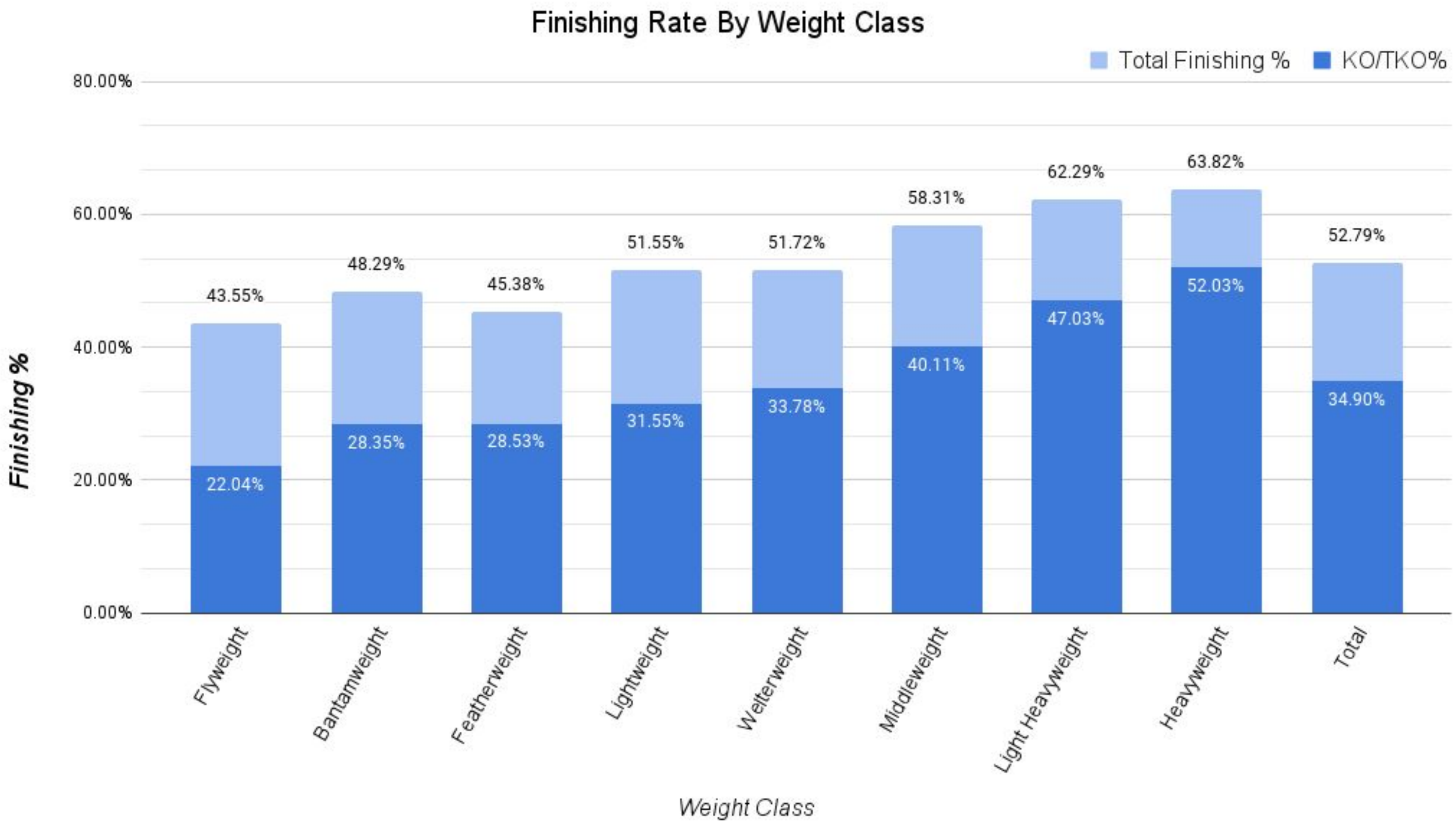
Weight Class	Wins by Favorite	Predicted Wins	Total Fights	Fav. Win %	xWin %	Win % Diff.
Flyweight	121	126	**186**	65.05%	67.74%	**2.69%**
Bantamweight	217	226	**321**	67.60%	70.40%	**2.80%**
Featherweight	228	237	**368**	61.96%	64.40%	**2.45%**
Lightweight	324	335	**485**	66.80%	69.07%	**2.27%**
Welterweight	333	343	**524**	63.55%	65.46%	**1.91%**
Middleweight	237	250	**379**	62.53%	65.96%	**3.43%**
Light Heavyweight	150	158	**236**	63.56%	66.95%	**3.39%**
Heavyweight	167	155	**246**	67.89%	63.01%	**-4.88%**
Total	***1777***	***1830***	***2745***	***64.74%***	***66.67%***	***1.93%***

Breakdown by Weight Class



Issues With Predicting Heavyweight

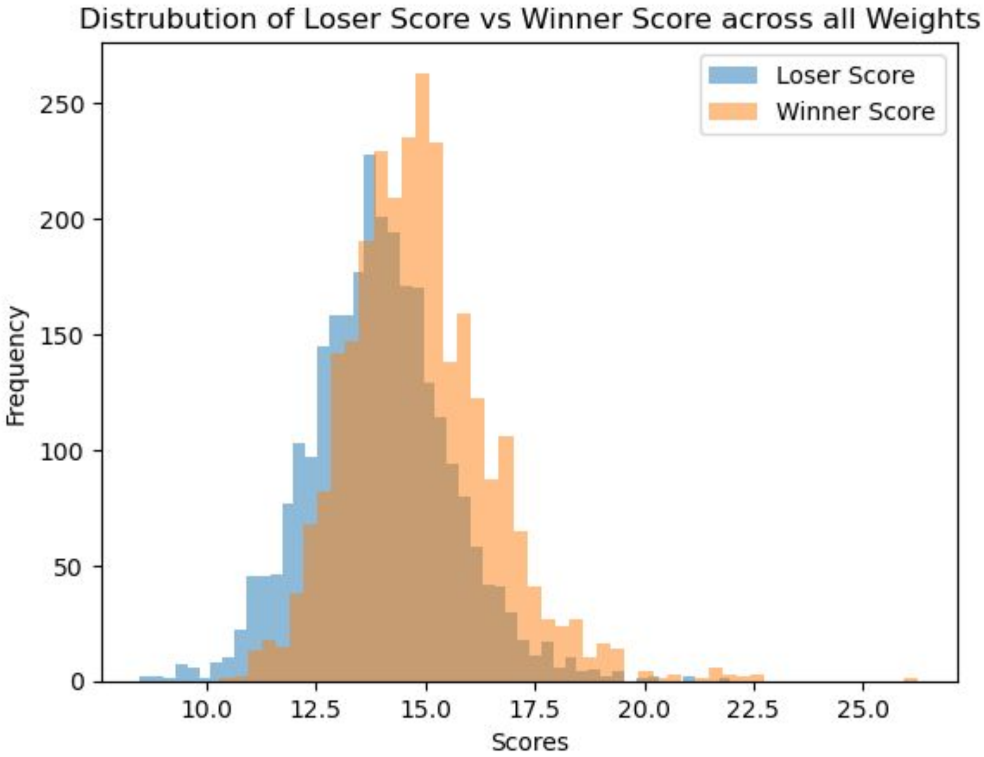
- As you may have noticed, Heavyweight was the only weight class where our model did not outperform the betting lines, and actually underperformed by 4.88%
- There were 2 major differences between Heavyweight and the other divisions that we could attribute this to:
    - First, Heavyweight fighters favored by betting lines had the highest win percentage out of any division (67.89%). However, the next closest division, Bantamweight, had a favored winning percentage that was only 0.29% lower, and our model outperformed betting lines by 2.8% there (Our 3rd highest margin), so we decided to investigate further
    - Next, we realized that that Heavyweight fights had the highest percentage of both knockouts and total finishes (KO + Sub.), with 52.03% of fights ending a knockout and 63.82% of fights ending in a finish, significantly higher than than the UFC average of 34.9% of fights ending in a knockout and 52.79% of fights ending in a finish
- We didn't want to compensate by giving much more weight to the favorite in the heavyweight fight, as we felt this would be the easy way out and contradicted the point of building this model in the first place



Breakdown of Finishing Rate by Weight Class

Fighter Score Distribution

- Note: Fighter Score is calculated differently for each weight class



Our Hi-Fidelity Interface Prototype

https://www.figma.com/proto/6QryVWlNOyrLZzaNcrpSGx/MMApredictHIFI?type=design&node-id=1-2&t=f1hnDIknLGBZWrHQ-1&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A2&mode=design

What We Would Do Different Next Time

- Some things we would expand upon if we had to redo this project include:
  - Expanding the formula for individual fighters beyond just an average of their past fighter score, perhaps by weighing more recent matches heavier than older ones, giving a boost to their score for wins by finish, boosting their score for wins over champions or other highly ranked opponents
  - Additionally, expanding the actual fight score formula itself to take into account things like the fighters rank, whether or not they are the champion, their record in their last 3 fights, etc.
  - The ultimate goal would be to follow through with creating a machine learning algorithm that could take all of these things into account and make real time, personalized adjustments based on the specific fighter