

# Predicting Mixed Martial Arts fights outcomes using Machine Learning

*[a Quantitative Betting Perspective]*

Maxime Rochkoulets – 23101510

Ryan Dempster – 20334796

# Motivation & Objectives

Is it possible to build a profitable model for betting on MMA (UFC) fights ?

----

Hypothesis: Bookmakers seem to sometimes give (too) high odds to some bets. Even if they are unlikely to occur, *since the odds are so high*, **the expected value of the bet could still be positive**, assuming our model gives us accurate/realistic probabilities.

----

The goal of this project is to answer **1.** Is it possible to build an accurate model for predicting MMA fights with the currently available data ? **2.** Can we use the probabilities given by this model to bet on the outcomes of fights ?

Let us take the example of the simplest type of bet ("*moneyline*"):

*"What fighter will win the fight ?"*

It can be seen as a random experiment with  $\Omega = \{ \mathbf{A}, \mathbf{B} \}$ , since here we assume that draws are impossible (they are not, but we will come back to that later).

Assume bookmakers (and the general opinion) think that fighter  $B$  is very unlikely to win, therefore the odds for this bet are **1.05** for "fighter  $A$  wins" and **7.50** for "fighter  $B$  wins".


Say our model predicts that fighter  $A$  will win with probability  $\mathbf{P(A) = 0.8}$ , which means that he thinks that fighter  $B$  has **20%** chances of winning.

We can compute the expected value of the random variable associated with the gain *if we bet on fighter  $B$* , giving us

$$\mathbb{E}(X_B) = \mathbb{P}(B)(O_B - 1) - \mathbb{P}(A) = 0.2 \times 6.5 - 0.8 = 0.5$$

It does not matter how unlikely this outcome is if the expected value is positive. But, at this point, *we don't know if such positive expected values exist, neither if our model will be accurate enough.*

# Dataset



REMY PEREIRA · UPDATED 6 MONTHS  
AGO

▲

14

Download (780 kB)

▼

⋮

## MMA Dataset 2023 (UFC)

Dataset of UFC fights, fighters, and fight stats from 1994-2023.

ufc\_event\_data.csv

ufc\_fight\_data.csv

ufc\_fight\_stat\_data.csv

ufc\_fighter\_data.csv

This dataset contains 4 different tables, but we only used `ufc_fight_data` and `ufc_fight_stat_data`.

fight_id	event_id	referee	f_1	f_2	winner	num_rounds	title_fight	weight_class	gender	result	result_details	finish_round
7218	664	Herb Dean	2976.0	2884.0	2884.0	5	F	Lightweight	M	KO/TKO	to \n Leg Injury	2
7217	664	Mark Smith	1662.0	2464.0	1662.0	3	F	Featherweight	M	Decision	Unanimous	3

...

fight_stat_id	fight_id	fighter_id	knockdowns	total_strikes_att	total_strikes_succ	sig_strikes_att	sig_strikes_succ	takedown_att
14436	7218	2976.0	0.0	34.0	19.0	32.0	18.0	0.0
14435	7218	2884.0	0.0	42.0	17.0	40.0	16.0	6.0

...

There was a lot of data pre-processing and cleaning required: and even if at first we had 7218 samples to work with...

# Training Data

After:

- 1) Removing rows with undefined values.
- 2) Keeping only three-rounds fights.
- 3) Removing female fights / weight categories where there are only few fights.
- 4) Dividing each in-fight stat by the duration of the fight.
- 5) Replacing each stat by the average of this stat in the last two fights.
- 6) Randomizing the position of the winner, since the UFC tends to put the fighter that is the favorite as fighter 1.

We are left with *2873 samples*, and our training data looks like:

knockdownsA	total_strikes_attA	total_strikes_succA	sig_strikes_attA	sig_strikes_succA	takedown_attA	takedown_succA	submission_attA	...
0.001544	0.182262	0.088415	0.175345	0.082486	0.000988	0.000988	0.000000	
0.035714	0.344762	0.121032	0.330317	0.108810	0.000556	0.000000	0.000000	

↑  
Features(28)

Targets →

winner	result	finish_round	finish_time
1	0	2	900
0	2	1	544
1	2	0	192

# Models

There are 3 dependent variables that we can try to predict: the winner of the fight, the way the fight will end or the round at which the fight will end.

We want to train algorithms that can give us probabilities as outputs (using functions such as *Sigmoid* or *SoftMax*), we used the following algorithms:

- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest

We did not use Decision Tree and Neural Network, because the former was giving very poor results after experimenting a bit, and the later because of the minimal amount of data we have.

# Results / Evaluation

*Random Forest* stands out as the most interesting model for us, since *we are not looking for the best accuracy* but for the model that can give us the most accurate probabilities.

This model also tends to take more risks, which can be interesting in our case.

**Diversity** is the number of times a model predicted a class that is not the dominant class in the dataset ÷ number of samples.

$$BS = \frac{1}{N} \sum_{t=1}^N \sum_{i=1}^R (f_{ti} - o_{ti})^2 \quad \leftarrow$$

<i>Winner</i>	Log. Reg.	Rand. Forest	SVM
Accuracy (%)	54.29	<b>54.63</b>	52.46
Log-Loss	0.693	<b>0.687</b>	0.692
Brier Score	0.498	<b>0.494</b>	0.499
Variance	0.464	<b>0.849</b>	0.384

<i>Result</i>	Log. Reg.	Rand. Forest	SVM
Accuracy (%)	<b>50.15</b>	49.14	49.94
Log-Loss	1.018	<b>1.009</b>	1.010
Brier Score	0.615	→ <b>0.471</b>	0.609
Variance	2.19	2.19	1.81
Diversity (%)	13.2	<b>16.4</b>	12.2

<i>Round</i>	Log. Reg.	Rand. Forest	SVM
Accuracy (%)	<b>57.04</b>	56.01	56.70
Log-Loss	0.982	0.997	<b>0.975</b>
Brier Score	0.581	→ <b>0.345</b>	0.581
Variance	3.14	3.25	2.83
Diversity (%)	2.63	<b>6.41</b>	0.22

# Problems

- Most fighters get their contract finished after 3 fights, lack of data if we want to take the average of the last 3, 4, 5 fights, that could yield a more precise model (less than 1000 samples).
- Draws are unlikely, *but not impossible* (similar to landing on green in roulette for example).
- When fights go to decision, the UFC seem to favor the fighter that "sells" the most.

# What's Next

- Maybe add physical attributes to the training data, for example, the height as  
 $\text{encoded\_height} = \text{height} - \text{average\_category\_height}$
- Find the best hyperparameters for Random Forest and maybe try XGBoost.
- Literature review / comparing our results with what has been done already.
- Reinforcement Learning ?
- Back-test our model using historical odds data and see if using this model for betting is profitable or not.



# Contributions

## Maxime:

- Main role – Implementation and Methodology; Evaluation
- Support role - Report

## Ryan:

- Main role – Evaluation; Report
- Support role - Implementation and Methodology