**MMA Write Up:**

These are just some interesting areas I highlighted in this project that others could improve on and just some ponderings I had as well, this in conjunction with code should give you a decent understanding of what this “product” is.

**N-Fight Average Approach:**

This function significantly improved the way I calculate a fighter’s N-Fight Average values compared to my previous method. The function takes a fighter's name and a specific date as input. Using these, it searches for all fight records involving that person (on either the A or B side), selects the most recent *N* fights, and retrieves the relevant performance data depending on which side the fighter was listed.

One limitation I’ve identified is that this approach doesn't currently account for the time span between fights. Ideally, instead of just requiring three fights, the function would check whether those fights occurred within a meaningful time window — for example, within the last year or 18 months. This would better represent a fighter’s current form. For instance, if a fighter competed once in 2010 and twice in 2020, averaging those values together might not yield a reliable prediction due to the long gap. In the future, I’d like to incorporate time-based filtering to exclude outdated records from the projection pool.

**Model Development:**

The initial version of this project was built using a logistic regression model in R. I exported the weights to Excel and manually ran predictions — a labor-intensive process. While not the most efficient start, it was a valuable learning experience, and I’ve since streamlined the process considerably.

Now, I use Python for model development, training, and testing. The model is designed to ingest values that represent future performance and compare those against historical fight outcomes. In the dataset, the "Outcome" column reflects whether Fighter\_A won (1) or Fighter\_B won (0), allowing for clear binary classification.

The promising part is that even with raw fight data, the model is performing at over 70% accuracy. While some variability in fight outcomes is inevitable — due to upsets, disqualifications, or overturned results — this accuracy suggests that the data does contain meaningful patterns. Given a consistent and reasonable way to project fighter outputs, we can use classifiers trained on historical outcomes to forecast future matchups with a fair degree of confidence.

**Fight Data Projections:**

This section could probably use a better name, but it focuses on deriving average values from a fighter’s recent bouts. This method is helpful for filtering out fights that don't fit well within the desired data intervals. Comparing actual fight data with these interval averages helps determine whether the interval is appropriate for predictive use.

Sample size is another factor. While more data is generally better, not every matchup will feature fighters with deep records. Some UFC fighters have fewer than 10 fights, and expanding too far back may introduce noise due to changes in age, opponents, rule sets, or era-specific trends.

Eventually, I plan to include visualizations to support this methodology — some of which are already in development — although I may refine them further to fit a consistent theme.

**Data Snapshots:**

A key question I grapple with is how to segment the data. As of April 2025, the dataset contains fewer than 10,000 rows. This relatively small size makes it challenging to build separate models for each weight class and gender, particularly since men's fights and middle-to-upper weight classes are overrepresented.

One idea was to include data from other MMA promotions to expand the dataset. However, this could dilute the model’s focus, similar to mixing NFL statistics with youth football — both involve the same game, but the context and skill levels are vastly different. While the idea is conceptually interesting, I’d need to be cautious with implementation.

When building training and testing sets, I experimented with N-Fight Averages ranging from two fights up to 10 or more. If the dataset were larger, it would make sense to filter out records that don’t meet the minimum number of fights required — this would help speed up the modeling process.

**Workflow:**

I plan to add a visual chart to clarify this section, so readers don’t have to sift through text like a recipe blog.

The process begins with the *Fight List* — a dataset of all past UFC outcomes. It includes key columns like "Outcome\_A," "Name\_A," "Name\_B," and "Date." These fields help determine winners, assign stats to each fighter, and timestamp the events. I update this dataset after major UFC events by scraping a reliable source. In practice, updates are done in bulk rather than after each event.

For upcoming events, I use a Python script to pull fight lineups. These are processed through a pipeline that calculates N-Fight Averages and checks which matchups meet model criteria. Those matchups are then passed into a classifier script, which outputs a fight-by-fight prediction.

Models are developed over time, often re-trained using the full dataset to create training, testing, and validation splits. The R&D process includes evaluating new model outputs, visualizing results, and validating performance on unseen data.

**Final Product:**

The final tool is straightforward: Given a list of fighters for an upcoming UFC event, the system filters eligible matchups based on model criteria, runs predictions using trained classifiers, and outputs a clear forecast of each fight's outcome.

The point is to give users — even those without deep MMA knowledge — a statistically grounded prediction. When I began this project, I was heavily invested in UFC news and rankings. Today, even with less engagement in the sport, I feel more confident in these model-driven predictions than I did relying solely on my prior knowledge.

Another goal is to provide a guide that others can follow — whether they're interested in applying this approach to other MMA organizations like ONE Championship or even adapting it to other sports with rich datasets. The methods here are flexible and can be repurposed wherever data and outcomes are clearly documented.

Singed: Robert C Schneider 05/04/2025