How accurately can UFC fight outcomes be predicted using historical performance data?

Machine Learning and Deep Learning (CDSCO2041C)

Somnath Mazumdar

Student: Tobias Fuchs, Alexander Seibel, Johannes Stärk

ID: 167609, 177126, 176206

Pages: 15

Characters (with Space): 34.537

Date: May 16, 2025

Copenhagen Business School
Department of Digitalization
2000 Frederiksberg
rg.digi@cbs.dk

Abstract

This project explores how accurately UFC fight outcomes can be predicted using historical performance data. Our central research question is: To what extent can UFC fight outcomes be predicted using historical performance data? To answer this, we developed a machine learning pipeline incorporating data pre-processing and feature engineering including lagged statistics and performance differentials. We evaluated logistic regression, random forest, XGBoost and deep neural networks on a cleaned and engineered dataset. XGBoost achieved the best performance, with 63% accuracy on the test set. Key predictive features included age difference, prior wins, knockdowns, and striking effectiveness. While the model provides valuable insights and highlights relevant patterns, the unpredictable nature of the combat sport Mixed Martial Arts, shaped by factors such as mindset, last-minute changes and injuries, limits the precision with which it can forecast outcomes. In conclusion, although machine learning can improve data-driven analysis in combat sports, expert judgement and contextual understanding should still be considered alongside it.

Keywords: UFC, Mixed Martial Arts, Sports Outcome Prediction, Machine Learning, XGBoost, Deep Neural Networks, Feature Engineering, Class Imbalance

Contents

1 Introduction

The UFC has evolved from a niche spectacle into a global sports powerhouse, attracting over 120 million followers on social media (Windsor, 2024). Its financial success is evident in the $1.3 billion revenue generated in 2023, with the majority coming from media rights and live events (Critchfield, 2024). This growth mirrors the expansion of the sports betting industry, which is projected to reach €86.4 billion by 2029 (Statista, 2025). Together, these trends highlight the increasing demand and value of accurate sports predictions.

Unlike many team sports, combat sports like MMA are shaped by a mix of physical attributes, fighting styles, and situational factors. This unpredictability makes outcomes harder to forecast, but also creates potential for data-driven modeling. Beyond betting, predictive models can support matchmaking, assess fighter performance, and enhance fan engagement.

As structured fight data becomes more accessible, machine learning provides new opportunities to analyze and predict outcomes. However, factors such as randomness, injuries, or psychological state still pose challenges. This project investigates whether machine learning models can meaningfully predict UFC fight results using historical pre-fight data.

2 Related Work & Research Questions

Predicting sports outcomes has been widely studied, particularly in team sports like basketball and football. These domains highlight both the potential and limitations of predictive modeling. For instance, Stekler and Klein achieved around 70% accuracy in basketball, though performance dropped in later tournament stages tournament due to increased competition (Stekler & Klein, 2012). In football, Beal et al. improved predictive accuracy by 6.9% over statistical baselines by incorporating contextual data, achieving 63.18% (Beal et al., 2020).

In contrast, MMA presents unique challenges due to its dynamic and individual nature. Several studies have explored UFC fight prediction. Turgut evaluated Random Forests and Neural Networks on UFC data, achieving moderate accuracies (59%) while emphasizing the need to prevent information leakage (Turgut, 2021). Similarly, McQuaide and McKinley tested multiple models, with Gradient Boosting performing best at 61.23% (McQuaide & McKinley, 2019).

Feature selection has also been explored. Apelgren and Eklund identified key predictors like age, reach, striking efficiency, and control time, while Turgut highlighted the importance of reach and win rate, despite high missingness in physical features.

Another challenge is to beat a naive model, that would achieve a relatively high accuracy by always predicting a win for the favored fighter 1.

Building on this prior work, our study seeks to examine whether structured historical data alone can provide reliable UFC fight predictions. We therefore ask:

- *Which machine learning model is best suited for UFC fight outcome predictions?*
- *Can a model outperform a naive baseline that always predicts a win for fighter 1?*
- *Which fighter characteristics most strongly influence outcomes?*

3 Conceptual Framework

Our UFC prediction pipeline follows five key steps. First, we perform data acquisition by collecting fight-level and fighter-specific data. In Data Preparation, we clean, merge, normalize, and develop features, including lagged averages and fighter differentials. Exploratory data analysis helps identify class imbalances and informs preprocessing. During Modeling, we train models of increasing complexity, from logistic regression to deep neural networks. Finally, Evaluation & Interpretation uses metrics such as accuracy, F1 score and ROC AUC to evaluate and explain model performance.

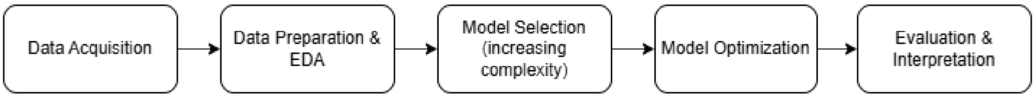
Data Acquisition
Data Preparation & EDA
Model Selection (increasing complexity)
Model Optimization
Evaluation & Interpretation

Figure 1: Modeling workflow for UFC fight outcome prediction

4 EDA and Preprocessing

4.1 Dataset Description

This project uses two primary datasets. The first, `df_fights`, contains all recorded UFC fights since 1994, including both pre-fight and post-fight information. Pre-fight data includes information like the event date, the names of both fighters or the weight class. Post-fight data includes the result of the fight, the method of victory, the number of rounds completed, and the official fight time. A major portion of the dataset consists of detailed statistics describing each fighter's performance during the fight. These include metrics like total takedown attempts or significant strikes landed. These statistics are reported both per round and in total, with an emphasis on significant strikes. The dataset contains 8,114 fights and 228 columns.

The second dataset, `df_fighters`, adds demographic and physical attributes (e.g., height, weight, reach) for each fighter.

As many columns of `df_fights` reflect post-fight results, they are not suitable for predicting fight results, as they would leak the fight outcome. For our modeling, we aim to use only pre-fight information, which we plan to receive by calculating one-fight-lagged averages.

4.2 Exploratory Data Analysis (EDA)

We began our exploratory data analysis by examining missing values. As expected, most of them appeared in round-based columns, since many fights end before later rounds occur.

The target variable `fighter1_result` shows a clear class imbalance: fighter 1 wins 5,096 times and loses 2,811 times. A naive model that always predicts a win would reach 64% accuracy, offering a baseline for evaluation (see Fig. 2). This imbalance is to be expected, given that Fighter 1 is usually assigned to the favourite in official UFC match listings.

Next, we looked at the fight lengths, which vary considerably (as seen in Fig. 3). This highlights the importance of normalizing dynamic fight statistics by the number of rounds, to ensure fair comparisons between shorter and longer fights.

We also identified 140 fights in which a fighter competed more than once on the same day, an uncommon scenario in modern UFC events. These instances could distort cumulative statistics and require special handling.

Moreover, around 2,100 fights involve at least one debuting fighter. As no prior stats exist for these individuals, lagged averages cannot be calculated. This makes imputation strategies essential in our preprocessing.

In summary, the datasets provide a strong foundation, with manageable missing values largely explained by fight structure. Based on these findings, we proceeded to clean the data, transform post-fight to pre-fight features, handle debut-related gaps, engineer experience-based features, and compute differences between fighter stats to support relative performance modeling.

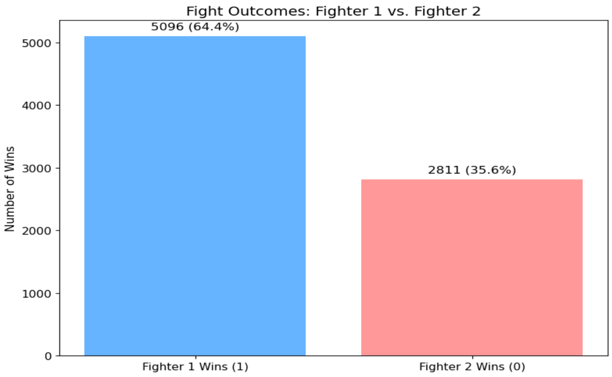

Figure 2: Class imbalance.

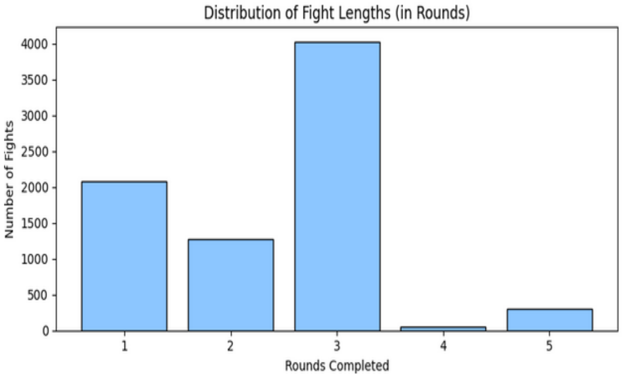

Figure 3: Distribution of fight lengths.

4.3 Preprocessing

Adding Fighter-Specific Information

Since `df_fights` lacked physical and demographic data, we merged it with `df_fighters`, which contains each fighter's weight, height, reach, stance, and date of birth.

Dropping Columns

First, we removed unnecessary columns such as `event_name` and percentage-based columns (e.g., "_pct"), as they are duplicated existing information. Also, we removed round-based statistics since they were too sparse due to early fight endings.

General Data Transformations

We cleaned and standardized the dataset to prepare it for modeling. This included formatting numeric and categorical variables and engineering relevant features. Weight classes were mapped to official UFC limits (ESPN, 2025), and the target variable was derived from whether `fighter1` won the fight.

Dropping Rows

To make the label binary, we removed all fights labeled "NC" (No Contest) or "D" (Draw), totaling 65. We also excluded 104 fights where a fighter competed multiple times on the same day, keeping only their first fight. To ensure consistent round-based stats, we dropped 161 fights with unusual `time_format` values, retaining only 3- and 5-round formats.

Handling Null Values

After cleaning, only 11 columns remained with missing data. Most had less than 1% missing values, except the reach of both fighters (3–10%). To preserve data, we decided to impute rather than drop rows. Missing `weight_limit_lbs` values were filled using a fighter's recent weight class. Other static features were imputed using the median within each weight class.

Normalization of Fight Stats

To account for varying fight lengths, we normalized all dynamic stats by the number of rounds. This ensures fair comparisons, as fighters with shorter fights otherwise appear to perform worse due to less accumulation time.

Computing One-Fight Lagged Averages

To convert post-fight stats into a pre-fight view, we computed one-fight-lagged averages of all dynamic stats, excluding static attributes. Cumulative victory method features (e.g., prior KO/TKO wins) were also created. Debut fights had missing lagged averages due to no prior data, so we applied two imputation strategies: zero-imputation and weight-class-level averages, yielding two datasets.

Adding New Features (Feature Engineering)

We engineered features capturing experience and fight patterns. These included total fight duration, cumulative and average rounds/time, total fights, wins, and title bouts. We also tracked each fighter's most recent weight class, win streaks, and days since last fight to model recent activity and momentum.

Building Differenced Datasets

To model relative strengths, we subtracted fighter2's values from fighter1's for each stat, avoiding ambiguity in feature ownership. This differencing approach helps the model learn competitive advantages. Non-fighter-specific columns (e.g., number of rounds, result) were retained.

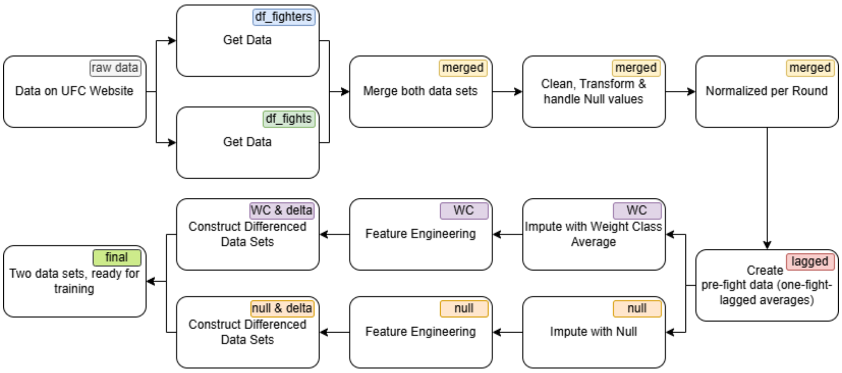

Figure 4: Data Preparation Process

5 Methodology

5.1 Train-Test Strategy

To evaluate the models properly and avoid data leakage, we applied two different train-test strategies both using stratified sampling due to the imbalance of the target labels.
For the classical machine learning models, we used a fixed 70 / 15 / 15 split for training, validation, and testing. In contrast, for training and fine-tuning deep neural networks, we applied 5-fold cross-validation. This approach is better suited for neural networks because they are more prone to overfitting and show greater variability in results due to random initialization. Since the dataset is relatively small, cross-validation helps make better use of the available data and provides more robust estimates.

5.2 Flipping Strategy for Data Augmentation and Class Balancing

As our dataset is small and imbalanced, we addressed this by augmenting the training data. Since all features are based on differences between fighter 1 and fighter 2, flipping the fighters reverses the feature signs and inverts the outcome. We used this property to create mirrored versions of each fight, but only in the training set. This helps the model learn to predict the winner based on actual performance differences rather than relying on the fixed position of the fighters. Because validation and test sets remain unflipped, we must interpret training and evaluation metrics carefully, as they are based on differently distributed data.

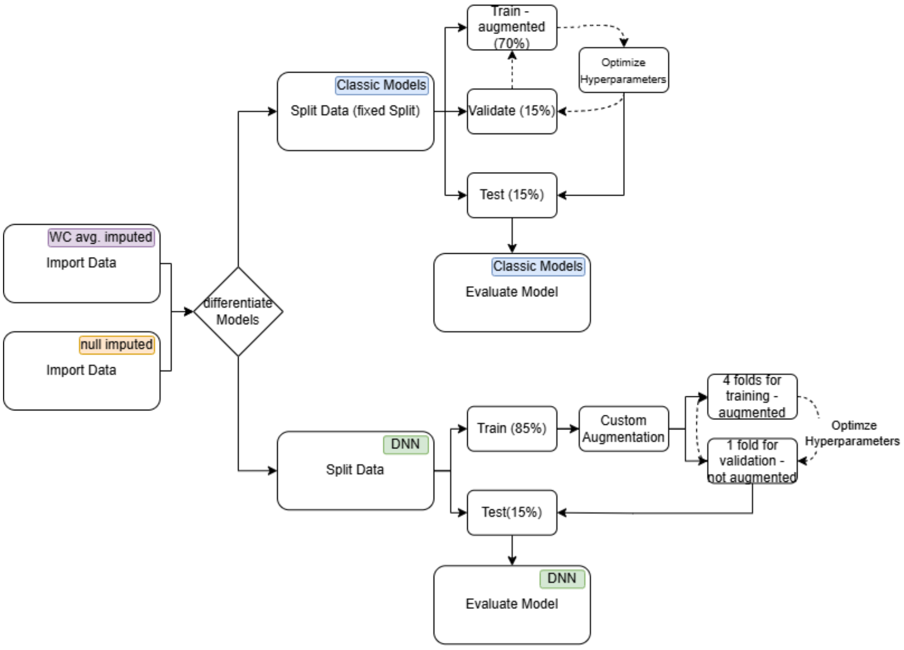

Figure 5: Train-Test Strategy combined with "flipping" for the different models

5.3 Evaluation Metrics

To evaluate the predictive performance of our models, we rely on three key metrics: Accuracy, F1 score, and ROC AUC. In our case, accuracy is a useful initial indicator of model performance, but it should be interpreted with caution. Since our validation and test sets are imbalanced, a model can achieve relatively high accuracy by simply predicting class 1 (fighter1 wins) most of the time, while failing to correctly predict class 0. This means accuracy alone may overestimate the model's true effectiveness. To address this, we also report the F1 score, as it balances the trade-off between false positives and false negatives. We compute F1 scores separately for each class to assess how well the model performs on both fighter wins (class 1) and losses (class 0). Finally, we include ROC AUC because it provides a more complete view of the model's ability to distinguish between classes. In our imbalanced setting, where accuracy and F1 scores can be influenced by class dominance, ROC AUC helps confirm whether the model is genuinely learning to separate wins from losses. It complements the other metrics by focusing on the overall quality of the model's predictions, rather than just the classification results.

5.4 Model selection & optimization

Logistic Regression, Random Forest & XGBoost

To predict UFC fight outcomes, we implemented a series of increasingly complex models, each optimized to fit the structure and challenges of our dataset.

We began with logistic regression, a linear model that served as an interpretable baseline. Its simplicity made it well-suited for initial experimentation, especially given our feature representation based on pairwise differences between fighters, which naturally aligns with linear decision boundaries (Géron, 2023, p. 144). To optimize the model, we manually tuned key hyperparameters using a fixed validation split. The regularization strength (C) was varied to find the optimal trade-off between underfitting and overfitting. The best results were achieved with strong regularization ($C = 0.01$), which helped prevent the model from overreacting to noise in our relatively small dataset. We selected L1 regularization to encourage sparsity in the learned coefficients, improving generalization by focusing only on the most informative features. The 'liblinear' solver was chosen for its compatibility with L1 penalties and stability on small to medium-sized datasets. Finally, we increased the iteration limit ('max_iter = 1000') to ensure convergence under the stricter regularization setting. This configuration produced a robust and interpretable baseline, setting a meaningful benchmark for more complex models.

Recognizing that linear models cannot capture the nonlinear interactions present in complex athletic performance, we introduced a random forest model. It is well-suited for structured data and handles mixed feature types effectively (Géron, 2023, p. 191). For optimization, we applied randomized search over a defined hyperparameter space, selecting configurations based on validation performance. The best model used 138 estimators and a maximum depth of 5, striking a balance between expressiveness and overfitting. To promote generalization and address class imbalance, we limited tree complexity, used a subset of features for splits, and applied class weighting and bootstrap sampling.

To further improve performance, we implemented XGBoost, a gradient boosting framework that constructs trees sequentially, with each one focusing on correcting the residuals of the previous (Géron, 2023, p. 201). Its strength in modeling subtle, nonlinear interactions made it particularly suitable for our task, where complex relationships between fighter attributes and past performance can influence the outcome. We performed a randomized hyperparameter search, selecting the best configuration based on validation F1 score to ensure balanced performance across classes.

The optimal model used a learning rate of 0.14 and 180 estimators, providing a good balance between training speed and predictive accuracy, which was helpful due to our limited dataset size. A low maximum tree depth of 3 was chosen to keep individual trees shallow, reducing the risk of overfitting to specific patterns in the training data. We applied column subsampling to ensure that each tree used only a random subset of features, introducing beneficial noise that improves generalization. Additionally, L1 ('reg_alpha = 0.58') and L2 ('reg_lambda = 0.96') regularization were included to penalize model complexity and further guard against overfitting. A small 'gamma' value (0.035) helped avoid unnecessary splits, which is especially important when working with engineered features that may contain redundancy.

Multilayer Perceptron (DNN)

Finally, we evaluated a (shallow) deep neural network to explore the benefits of more expressive, nonlinear modeling. As our dataset contains tabular, numerical features, we decided to go with a shallow MLP using binary cross-entropy as the loss function to predict binary outcomes. Given the dataset size (approx. 10,000 rows, 55 features), a deep neural network with 5 or more hidden layers would risk overfitting.
Thus, as a baseline we started with 3 hidden layers with 128 neurons and a learning rate of 0.01. However, the model shows limited learning capacity. Both training and validation accuracy hover around 50%, with ROC AUC close to 0.5, indicating the model is barely better than random guessing. The F1 scores for class 0 are particularly low, suggesting the model struggles to recognize when fighter 1 is expected to lose. While class 1 recall is relatively high, this comes at the cost of poor precision and imbalance across metrics. These results strongly suggest underfitting.
So as a next step, we tried a model with less capacity (less layers and less neurons). The model significantly improved in training and validation accuracy, but the gap increased at the same time. This is expected to some degree since the training set is balanced (due to flipped rows), while the validation set is skewed toward one class. However, the size of the gap suggests the model isn't learning to generalize well. Also, it still shows a clear imbalance in how it handles the two outcome classes. The model performs significantly better on predicting wins for fighter 1 (class 1) than losses (class 0). This can be seen in the large difference between F1 scores across classes, both during training and validation. This indicates that the model is still underfitting.
As a next step, we tried a model with a lower learning rate of 0.001. This led to a noticeable improvement in model performance. Both training and validation accuracy increased, and the gap between them narrowed, suggesting better generalization and less overfitting. More importantly, the model now performs more reliably on both classes. The F1 score for class 0 (fighter 1 loses) increased significantly, indicating that the model has started to learn to identify the minority class more effectively. Meanwhile, performance on class 1 remained stable, meaning the improvement didn't come at the cost of majority class accuracy. The higher ROC AUC further supports this improvement. Through a grid search, we explored different combinations of learning rates and model capacities within a range that seemed reasonable to us. The best-performing model in terms of training accuracy and class balance used 2 hidden layers, 128 neurons, and a reduced learning rate of 0.0005. While this configuration achieved a marginal improvement in training accuracy, the overall performance gains were minimal. Validation metrics such as accuracy, F1 scores, and ROC AUC remained nearly unchanged, with only slight increases in class 0 recall and F1, suggesting a modest improvement in capturing the minority class. However, these changes are within the expected noise margin and do not indicate meaningful progress.

Further fine-tuning, such as applying learning rate schedulers, did not yield meaningful improvements and only increased model complexity without enhancing performance. We also chose not to apply additional regularization techniques such as L1/L2 penalties or dropout, as the network was relatively shallow and training curves showed no clear signs of overfitting. Moreover, our training set was augmented through row flipping to improve class balance, but the overall dataset size remained modest, making heavy regularization unnecessary and potentially harmful to learning subtle patterns.

Model	**Split**	**Accuracy**	**AUC-ROC**	**Precision**		**Recall**		**F1-Score**	
				Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
1. Baseline DNN	Train	0.50	0.50	0.20	0.30	0.40	0.60	0.27	0.40
	Val	0.53	0.50	0.14	0.34	0.40	0.60	0.21	0.47
2. DNN (less capacity)	Train	0.55	0.58	0.37	0.55	0.30	0.81	0.33	0.64
	Val	0.62	0.57	0.27	0.67	0.27	0.81	0.27	0.73
3. DNN (lower lr)	Train	0.58	0.63	0.63	0.56	0.49	0.76	0.49	0.65
	Val	0.63	0.62	0.48	0.69	0.37	0.77	0.41	0.73
4. DNN (Grid Search)	Train	0.59	0.64	0.64	0.57	0.43	0.75	0.51	0.65
	Val	0.63	0.62	0.48	0.69	0.39	0.77	0.43	0.73

Table 1: performance metrics showing the process of finetuning the DNN on the validation set.

When running the fine-tuned model on the test data, its uncertainty becomes clear. As shown in Figure 6, most predicted probabilities lie between 0.4 and 0.7, with significant overlap between the classes. This indicates the model struggles to confidently distinguish wins from losses.

To address this, we explored threshold tuning. Figure 7 shows that increasing the threshold improves the F1 score for class 0 (fighter 1 loses), which is typically harder to predict. However, this comes at the cost of overall accuracy and a drop in class 1 performance. Since the test data is imbalanced, accuracy is less reliable as a metric. Prioritizing class 0 via a higher threshold ($\tilde{0}$.65) may lead to more balanced and practically useful predictions. However, for our final comparison with the classic models, we chose to not use threshold tuning as it would introduce an additional layer of optimization specific to the neural network, making the comparison less fair and less consistent across model types.

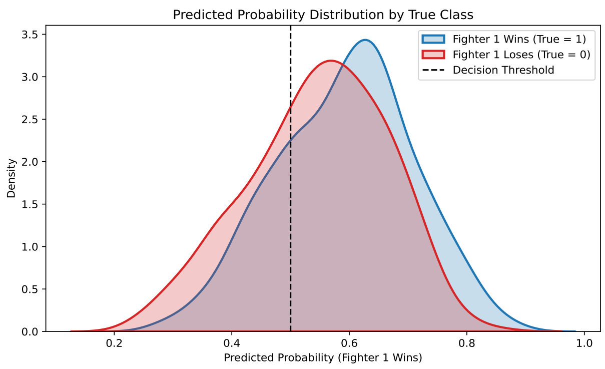

Figure 6: Predicted probability distributions.

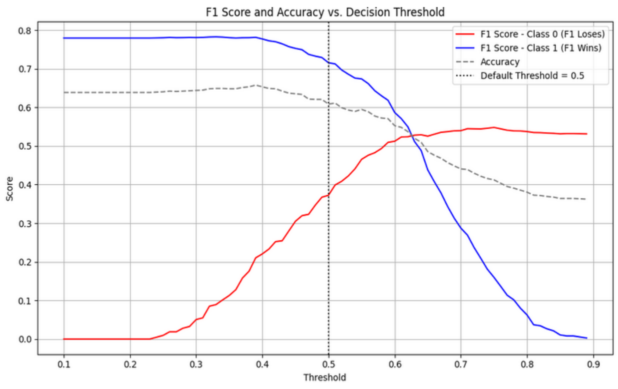

Figure 7: F1 score vs. threshold plot.

6 Results

Table 2 shows the four optimized models. Each model was trained using the same balanced dataset and evaluated using a fixed imbalanced test set. Although two imputation strategies were initially introduced, only the weight class average method is reported here, as both variants produced similar results and this method aligns better with domain intuition.

XGBoost achieved the highest overall accuracy (0.63) and AUC-ROC (0.63) among all models, along with balanced class-wise F1 scores (0.50 for class 0 and 0.71 for class 1). These results suggest relatively strong, consistent classification performance, particularly in distinguishing wins from losses, independent of class distribution.

Despite its simplicity, logistic regression performed competitively, achieving an accuracy of 0.62 and an F1-score of 0.69 for class 1. However, its lower recall for class 0 (0.55) indicates that the linear model has difficulty correctly identifying fighter 1 losses, reflecting its limited ability to model complex decision boundaries.

Random forest slightly improves recall for class 0 compared to logistic regression, but its performance is similar overall. Its precision and F1 scores suggest that it captures more complex relationships, yet it still underperforms XGBoost in most categories.

The deep neural network achieves the highest F1-score for class 1 (0.72), matching the performance of XGBoost in this category. However, its ability to detect class 0 cases (fighter 1 losses) is limited, with a recall of just 0.33 and an F1-score of 0.38. This indicates that while the DNN is good at correctly predicting wins, it struggles significantly with identifying losses. Its relatively low precision and recall for class 0 suggest that the model leans heavily toward the majority class, despite training on a balanced dataset.

In addition to standard performance metrics such as accuracy, F1 score, and ROC AUC, it is usually important to consider model complexity and resource requirements when comparing different approaches. However, since our dataset is relatively small and the neural network used was shallow, model complexity had little impact on training time or computational demands. Classical models like logistic regression, random forest, and XGBoost also trained efficiently due to the limited size and dimensionality of the data. As time and computing power were not limiting factors in our case, we found that runtime differences were negligible and did not require further analysis.

In summary, XGBoost offers the most balanced and reliable performance across all key metrics, making it the strongest representative of traditional models. The DNN shows potential for capturing underrepresented patterns, but it requires more tuning or data to outperform tree-based methods consistently.

Model	Accuracy	AUC-ROC	Precision		Recall		F1-Score	
			Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Logistic Regression (Baseline)	0.62	0.62	0.48	0.72	0.55	0.66	0.51	0.69
Random Forest	0.61	0.62	0.47	0.72	0.56	0.64	0.51	0.68
XGBoost	0.63	0.63	0.49	0.72	0.52	0.69	0.50	0.71
Deep Neural Network	0.61	0.60	0.45	0.67	0.33	0.78	0.38	0.72

Table 2: Performance comparison of optimized models on the test set.

7 Discussion & Interpretation

In this section, we connect the evaluation results of our models to the central research questions posed earlier. We aim to interpret where and why certain models perform better, what limitations they face, and how their predictions align with the structure and dynamics of UFC fight data.

Which machine learning model is best suited for UFC fight outcome predictions?

Our results show that XGBoost performs best overall. It achieved the highest accuracy (0.63), the best AUC-ROC score (0.63), and balanced F1-scores for both classes. This suggests that XGBoost is the most effective model in capturing the structure of the data and making consistent predictions across different fight outcomes.

However, accuracy is not the only metric to consider. Because fighter 1 wins around 64% of the time, a model that mostly predicts wins can reach high accuracy without truly understanding the task. That's why we also look at class-wise F1-scores, precision, and recall.

XGBoost offers a good balance between the two classes, but other models show different strengths. The deep neural network, for instance, reaches the highest F1-score for class 1 (fighter 1 wins) and performs well on recall for that class. While it does not outperform XGBoost overall, it could be useful when the goal is to detect the more common outcome with higher confidence.

Logistic regression and random forest also performed competitively. Logistic regression, despite its simplicity, achieved an accuracy of 0.62 and a solid F1-score for class 1. This shows that even a linear model can work well when given the right inputs. Random forest showed similar results, slightly improving class 0 recall but not surpassing XGBoost.

In summary, if the goal is balanced and accurate predictions, XGBoost is the best choice. If the focus shifts to interpretability or specific class performance, simpler models or the DNN can still provide useful alternatives.

Can a model outperform a naive baseline that always predicts fighter 1 to win?

This question is important because of the class imbalance in our dataset. It might seem that our models, even after careful tuning, fail to outperform a naive model that simply always predicts a win for fighter 1. Such a model would achieve an accuracy of around 64%, which is already higher than some of our trained models. At first glance, this appears strong and hard to beat.

However, accuracy alone can be misleading, especially in imbalanced datasets like ours. The naive model only makes one type of prediction and completely ignores fighter 1 losses (class 0). As a result, its F1-score for class 0 is zero, meaning it fails to capture one half of the task. In contrast, our models produce non-zero F1-scores for both classes, indicating they make predictions for both outcomes and not just the majority class.

Another important point is that the naive model has an AUC-ROC of 0.5, which is the same as random guessing. Our trained models consistently reach AUC scores above 0.6. This metric reflects how well a model separates the two classes across all thresholds, and a higher score shows the model has learned useful patterns beyond the base rate.

Even the deep neural network, which has slightly lower accuracy than the naive baseline, offers more balanced performance. It identifies losses more often, which is valuable in real-world scenarios like betting or risk analysis, where predicting rare but impactful outcomes matters.

So while the naive model sets a high benchmark for accuracy, it does so by ignoring one class entirely. Our models go beyond that. They recognize both outcomes and provide more meaningful predictions, something accuracy alone doesn't capture.

Which fighter characteristics most strongly influence outcomes?

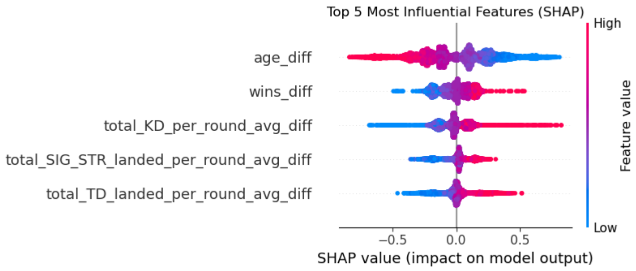

Figure 8: Top 5 Features Driving Model Predictions (SHAP Analysis)

The SHAP (SHapley Additive exPlanations) summary plot in Figure 8 shows the most influential features used by the XGBoost model. SHAP values, based on cooperative game theory, quantify how much each feature contributes, positively or negatively, to a prediction (GeeksforGeeks, 2024). Each point represents a single prediction, with the x-axis showing impact on model output and color indicating feature value (red = high, blue = low).

The plot highlights the five most impactful features. The top feature is `age_diff`, where younger fighters (more negative values) are linked to a higher win probability, shown by red points on the right. Next is `wins_diff`, indicating fighters with more prior wins increase the model's confidence. Technical performance indicators such as knockdowns, significant strikes, and takedowns per round also rank highly, emphasizing the role of past striking and grappling success in predicting outcomes.

A key strength of XGBoost is its interpretability. Unlike deep neural networks, which often act as "black boxes", XGBoost supports post hoc explanation via SHAP analysis. This transparency improves understanding of which fighter traits drive predictions, making the model more trustworthy and valuable for practical use in sports analytics.

7.1 Ethical Considerations

Our model raises key ethical concerns, particularly around fairness and transparency, two core principles of responsible AI (Microsoft, 2024). A major fairness issue is the strong gender imbalance in the dataset, with male fighters vastly overrepresented, likely leading to better model performance for men than for women. In high-stakes contexts like betting, predictive models also raise concerns about addiction and exploitation, as algorithmic outputs can reinforce risky behaviour and disproportionately impact vulnerable individuals.

Moreover, complex models such as deep neural networks offer only limited transparency by nature, making it difficult to trace feature influence or detect biases and errors in their reasoning.

7.2 Limitations

The limitations are due to the nature and structure of the dataset, but also to the nature and unpredictability of the sport itself.

Missing values were imputed using class medians, or zeros and weight-class averages for debut fights. While these methods are practical, they introduce bias by either generalising or underestimating individual performance. As debut fights account for around 25 % of the data, this restricts the reliability and accuracy of the model for key traits. In addition, key factors such as training quality, injuries, or mindset are missing from the data, reducing the accuracy of the prediction. Furthermore, the male-dominated dataset limits generalizability to women's fights. Competitive sports are inherently unpredictable, and MMA takes this to the extreme, where a single strike, referee call, or tactical surprise can decide the outcome. This volatility makes accurate forecasting from historical data particularly challenging.

8 Future Work & Conclusion

Future Work

For future work, adding external context features, such as betting odds, could increase predictive accuracy, as they reflect expert insight and public sentiment, factors that have been shown to be valuable in sports analytics. For example, Hubáček et al. found that betting odds significantly improved model performance in predicting football match outcomes (Hubáček et al., 2019). Another way to expand the dataset is by incorporating text-based features using models like BERT, which can extract context from interviews or pre-fight statements to capture a fighter's mindset or public hype. These signals can enrich statistical features and improve predictions.
A key challenge arises when fighters have very similar stats, e.g., reach, experience, or recent form, making it difficult for models to distinguish between them. Whereas our approach uses static and lagged averages to represent performance, it lacks the ability to capture more slight time-based trends or complex relationships between features. MambaNet takes a different approach by incorporating more dynamic and context-aware representations, which can better handle such closely matched scenarios (Zhou et al., 2020). Applying this kind of architecture to UFC data could improve prediction accuracy, especially when historical information is sparse or fighters appear statistically similar.

Conclusion

In conclusion, our project demonstrates that predicting UFC fight outcomes based on historical performance data is feasible, though constrained by the unpredictable nature of the sport. Among the models tested, XGBoost achieved the best results with 63% accuracy and balanced class-wise performance, making it the most suitable choice for structured prediction. Key predictive features included age difference, prior win records, and technical performance indicators like knockdowns, significant strikes, and takedowns.
A key part of our analysis was comparing these models against a naive baseline that always predicts a win for Fighter 1. While this baseline achieves 64% accuracy due to the class imbalance in the data, it entirely fails to recognize losses and offers no insight into rare or unexpected outcomes. In contrast, our trained models provide more nuanced predictions, accounting for both victory and defeat, and demonstrating learned structure beyond the base rate.
Despite some limitations, particularly in handling debut fighters with no prior data, the results highlight the potential of machine learning for structured fight prediction. Still, such models should be viewed as decision-support tools rather than definitive predictors, particularly given the ethical implications surrounding fairness, representation, and their application in betting contexts.

References

Apelgren, S., & Eklund, C. (2024). Predicting UFC matches using regression models.

Beal, R., Middleton, S., Norman, T., & Ramchurn, S. D. (2020). Combining Machine Learning and Human Experts to Predict Match Outcomes in football: A Baseline Model.

Critchfield, T. (2024, February). Earnings Report: UFC Generated $1.3 Billion in Revenue for 2023. Retrieved May 4, 2025, from //www.sherdog.com/news/news/Earnings-Report-UFC-Generated-3613-Billion-in-Revenue-for-2023-192772

ESPN. (2025, September). Current and all-time UFC champions. Retrieved May 8, 2025, from https://www.espn.com/watch/syndicatedplayer

GeeksforGeeks. (2024, April). SHAP : A Comprehensive Guide to SHapley Additive exPlanations [Section: Machine Learning]. Retrieved May 14, 2025, from https://www.geeksforgeeks.org/shap-a-comprehensive-guide-to-shapley-additive-explanations/

Géron, A. (2023). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (Third edition). O'Reilly.

Hubáček, O., Šourek, G., & Železný, F. (2019). Exploiting bookmaker odds for better prediction of football matches. *Knowledge-Based Systems, 188.*

McQuaide & McKinley. (2019). Applying Machine Learning Algorithms to predict UFC Fight Outcomes. *Stanford University.*

Microsoft. (2024, September). What is Responsible AI? https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai

Statista. (2025, January). Sports Betting - Weltweit | Statista Marktprognose. Retrieved May 4, 2025, from http://frontend.xmo.prod.aws.statista.com/outlook/amo/gambling/sports-betting/weltweit?currency=EUR

Stekler, H., & Klein, A. (2012). Combining Machine Learning and Human Experts to Predict Match Outcomes in football: A Baseline Model.

Turgut, M. (2021). Machine Learning approach to predicting Mixed Martial Arts matches. https://drewhendrickson.github.io/theses/F2020_Mehmet_Turgut_thesis.pdf

Windsor, A. (2024, May). UFC Viewership Statistics 2025 | Latest UFC Trends & Insights. Retrieved May 4, 2025, from https://minimumdepositbettingsites.com/guides/statistics/ufc-viewership/

Zhou, Y., Zhang, L., & Yang, H. (2020). MambaNet: A hybrid neural network for predicting the NBA playoffs. *Proceedings of the AAAI Conference on Artificial Intelligence.*