Ultimate Fighting Championship Dataset Analysis

Mallory Go[1]
[1]Department of Epidemiology, Brown University School of Public Health

Github repository
https://github.com/mallory-go/final.git

Introduction
The Ultimate Fighting Championship (UFC) is a world wide mixed martial arts (MMA) organization representing thousands of international fighters. The dataset was a publicly available dataset of all UFC fights from the mid-2010s, to December 7, 2024, scraped from ufcstats.com, an independent site that provides comprehensive fight statistics. The data is sourced from FightMetric, the system the official UFC website and broadcast uses for its own records (Ultimate UFC Dataset). The dataset contains three data files describing the event (location, title fight, winners, etc), fight (strikes landed, takedowns, etc), and fighters (biography details, overall performance of the fighter, fighter record, etc). The attributes for each fighter were inputted into the database as the most recent data on the fighter.

Purpose
This analysis aimed to predict fight wins – with a particular focus on title fight wins – using machine learning algorithms. I predict that all models will not have accurate performance (AUROC > 0.7) due to the volatility of the sport and – historically – unobserved or unrecorded factors such as injuries, psychological issues, or even pure luck (as seen by "upsets" or unexpected outcomes) can impact or determine the winner of a fight. The Red corner is generally considered the "favorite" fighter, the fighter predicted to win the fight, and the Blue corner is assigned to the other fighter. The UFC organization has not disclosed how fighter odds or the 'favorite' fighter is determined (Exploring Patterns and Predictors in UFC Fight Outcomes).

Target variable and Features
The Ultimate Fighting Championship Dataset is a dataset created for fight analysis, with fighter attributes and fight outcomes. The target variable is 'Winner' or the fighter that won a particular fight. AUnlike other datasets, each row of this dataset represented one fight and only the statistics of the fighters at the time of the fight. 'Winner' is coded as 'Red' ('1' after preprocessing) for the fighter in the red corner – the UFC designated favorite – or Blue ('0' after preprocessing) for the fighter in the blue corner. There are 6528 datapoints and 235 feature columns in the original dataset. The feature columns consist of event or fight details (location, date, etc.) and fighter characteristics and history (win streak, stance, citizenship, etc). To collapse the amount of features, I calculated the difference between 'Red' and 'Blue' measurements. For example, reach_diff would be the reach (arm length) of the 'Blue' fighter subtracted from the 'Red' fighter.

Exploratory Data Analysis (EDA)

I used a bar chart to check the distribution and balance of the target variable, 'Winner'. Each fight has a 'Red' or 'Blue' winner.

Figure 1. Distribution of target variable

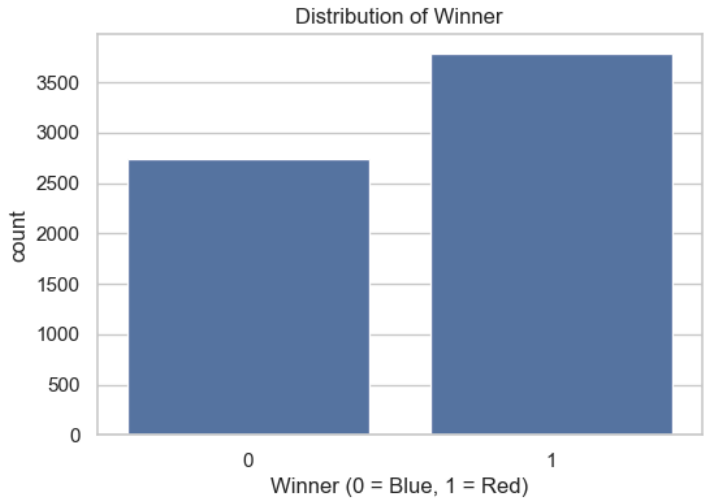
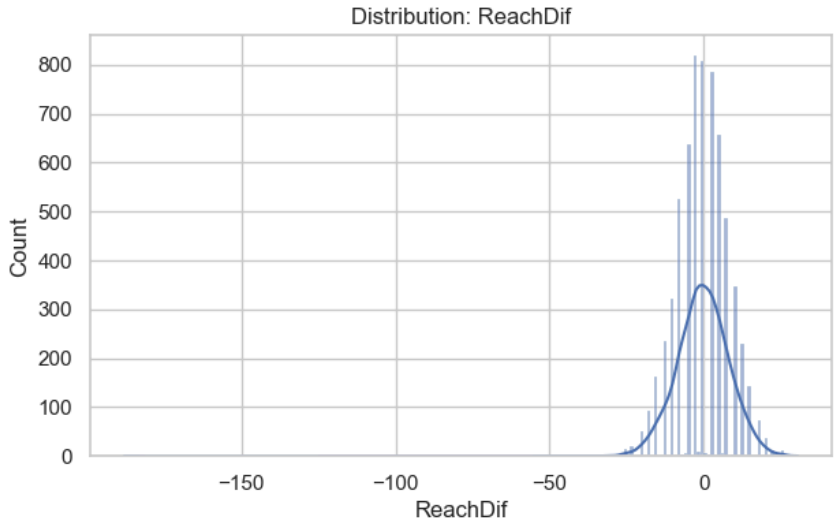


Figure 2. Distribution of Fighter reach in inches



Methods

Data preprocessing

All fighter-level variables were combined into a structured dataset where the outcome variable was Winner (binary: Red vs. Blue). To prevent label leakage and ensure model fairness, all features containing “Blue” or “blue” were removed, resulting in a “Red-only” feature matrix that does not implicitly encode the outcome.

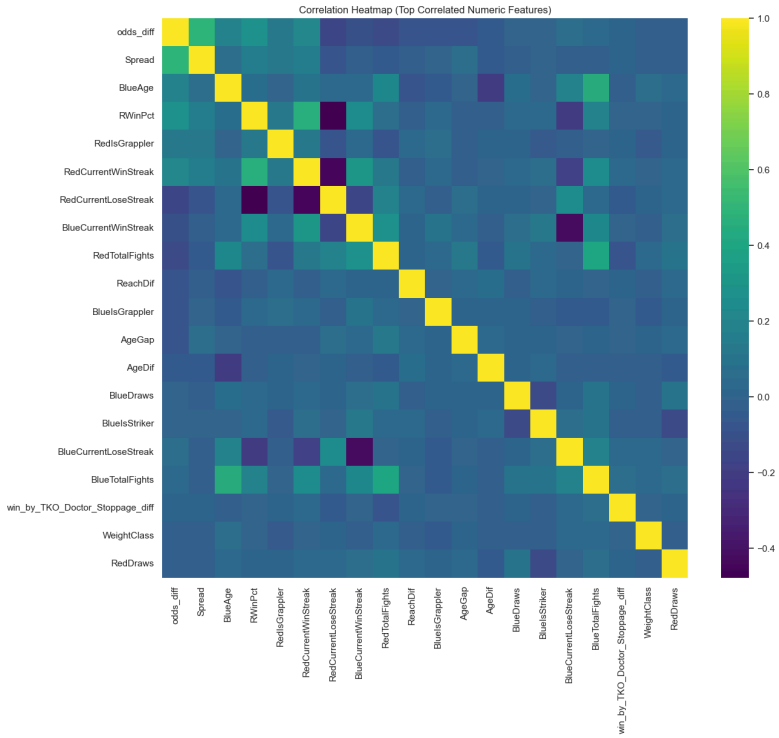
For missing data, I checked the percentage of missing data for all features (per column) and used OneHotEncoder for categorical features (imputed with the token “missing”). Columns with potential data leakage were also dropped (features with information on how the fight was won). As the original features

contained strings, they needed to be transformed into numbers using OneHotEncoder. Numeric features were standardized using StandardScaler. All preprocessing steps were implemented using a unified scikit-learn ColumnTransformer pipeline to ensure consistent transformations across all models.

Columns with a proportion of missing over 98% were excluded from the analysis. For feature reduction, we used a correlation matrix and calculated the Pearson correlation coefficient. We dropped strongly correlated numeric features ($|r| > 0.50$) with the strategy that for each pair with $|r|>0.5$ drop the feature with lower variance (Akoglu, 2018).

The dataset was split into train, validation, and test subsets (6:2:2) with cross validation using 10 KFold Split. The split and cross-validation method was chosen due to the balanced dataset and to reduce variance in validation metrics. Using both a fixed train/validation/test split and K-fold cross-validation allows us to measure uncertainty from two sources.

Figure 3. Correlation matrix of numeric variables after feature reduction



ML pipeline

The following machine learning algorithms were used for the analysis: Logistic Regression (LR), Support vector machines (SVM), Random forest (RF), XGBoost classifier algorithm. The model algorithms were assigned five different random states for splitting – with the exception of XGBoost. We tuned each model using GridSearchCV with 5-fold stratified cross-validation and for every algorithm we defined a grid of candidate hyperparameters and allowed the grid search to evaluate all parameter combinations. The best parameters were selected automatically using the highest average cross-validated accuracy and were extracted using grid.best_params_ and stored in the results summary table. The parameters differed across models (e.g., RandomForest preferred shallower trees while XGBoost

preferred smaller learning rates). The model performance algorithm was the area under the receiver operating curve (AUROC). Global interpretability was computed using permutation importance and model-specific feature importance (weight, gain, cover), while local interpretability was measured using Shapley Additive exPlanations (SHAP) values.

Table 1. Hyperparameter values by machine learning algorithm

Algorithm	Best Parameters	Parameters	Mean Test Score	Std Dev	Best Validation Score	Test Score
Logistic Regression	`'C': 0.01`	`{'classifier__C': [0.0001, 0.001, 0.01, 0.1]}`	0.650	<0.001	0.6589	0.650
Random Forest	`'max depth': 10, 'Minimum samples to split a node': 5, 'number of trees': 300`	`{'classifier__n_estimators': [100, 300], 'classifier__max_depth': [None, 10, 20], 'classifier__min_samples_split': [5,10]}`	0.648	0.003	0.650	0.649
SVM	`'C': 0.1, 'kernel': 'linear'`	`{'classifier__C': [0.001, 0.01, 0.1], 'classifier__kernel': ['rbf', 'linear']}`	0.653	<0.001	0.658	0.653
XGBoost	`'boosting size': 0.01, 'max depth': 3, '# of boosting rounds': 300`	`{'classifier__n_estimators': [100, 300], 'classifier__max_depth': [1, 3], 'classifier__learning_rate': [0.001, 0.01]}`	0.659	<0.001	0.657	0.659

Baseline accuracy was 0.5947 while the best performing model had an accuracy of 0.6590. Across all evaluated models, performance clustered in the mid-60% range, with XGBoost finishing as the best model, as seen by both the test accuracy and cross-validated AUROC. The optimal XGBoost configuration used a learning rate of 0.01, max_depth of 3, and 300 estimators, indicating that relatively shallow, strongly regularized trees best captured the structure in the data (Table 1). Simpler models such as Logistic Regression and SVMs also performed competitively, suggesting that the underlying relationships between fighter statistics and match outcomes are largely linear and/or relatively low-complexity (Table 1). Some models show a test-accuracy standard deviation of less than 0.0001 across random seeds. In practice, this indicates that the model's performance is extremely stable across different initial random states, with accuracy variations below 0.001. In terms of interpretability, the

models consistently identified betting odds difference (odds_diff) as the strongest predictor, with secondary contributions from physical advantages such as reach, age, and striking accuracy. Overall, while the models outperform the naive baseline from fighter statistics, I infer that the inherent unpredictability of UFC events combined with missing qualitative and temporal information likely limits accuracy to approximately the 66% ceiling using publicly-available tabular data alone (Exploring Patterns and Predictors in UFC Fight Outcomes).

Figure 4. Confusion matrix for XGBoost algorithm

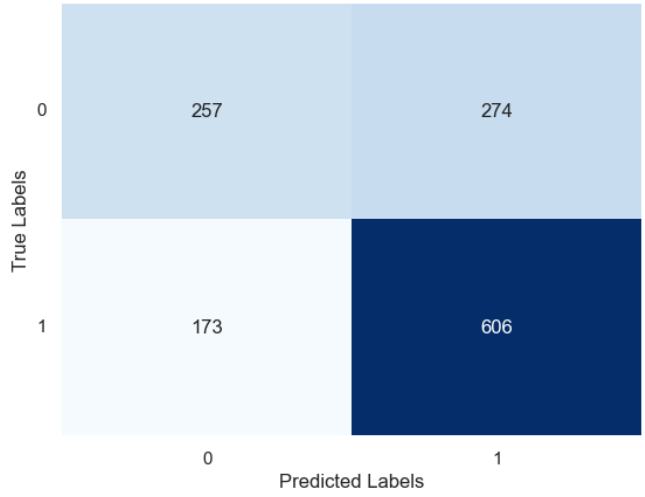


Figure 5. Model mean test accuracy by algorithm

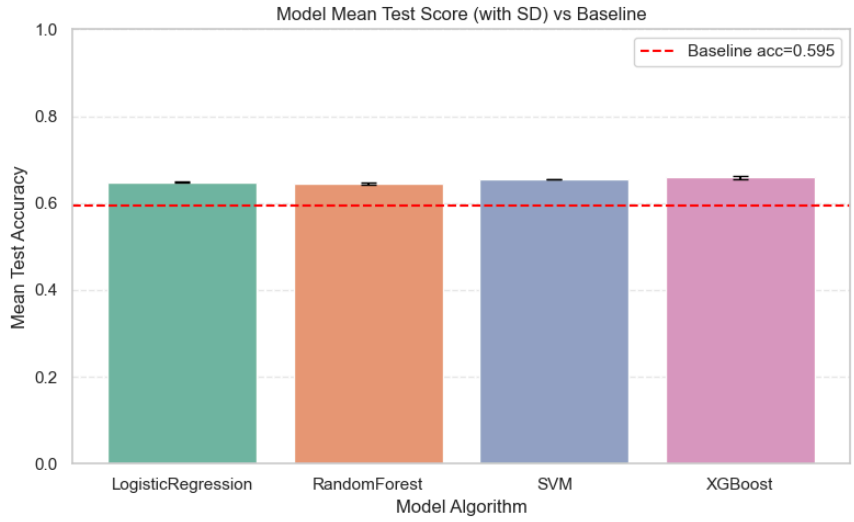
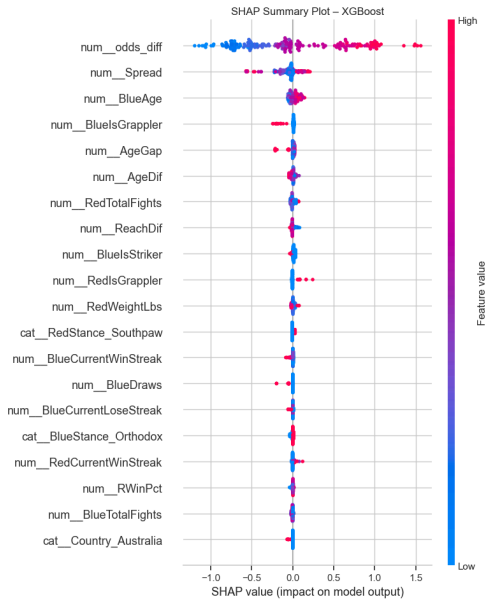


Figure 6. SHAP summary plot for XGBoost (best performing model)



Three different global feature importances for best model: XGBoost

Three metrics were used to estimate feature importance for the best-performing model (XGBoost): permutation importance, model-specific XGBoost importances (gain, cover, and weight), and SHAP (Shapley Additive Explanations) summary values. Although these methods differ in how they quantify importance, all three consistently identified odds difference (odds_diff) as a dominant predictor in the dataset. Secondary predictors such as reach and age differences, stance interactions, grappling/striking interactions, and experience metrics contributed but remained less impactful. SHAP additionally clarified the direction of effects, showing that more favorable betting odds strongly increase predicted win probability. While the models outperform a naive baseline, performance stabilizes around 65–68% accuracy, suggesting that publicly available fighter statistics capture only part of the true variability in fight outcomes, with unobserved factors such as training quality, past injuries, stylistic nuances, psychological elements, and history of flukes likely setting an upper limit on model predictability (Exploring Patterns and Predictors in UFC Fight Outcomes).

Outlook

Model performance could be improved by expanding or diversifying the features and dataset. Specifically, increasing the sample size, incorporating more recent fight data, adding time-series features such as momentum, layoffs, or changes in training camps, and including physiological metrics or measures of betting-market volatility could enhance predictive capacity (J. McDermott, 2025). These additions would help capture dynamic factors that influence fight outcomes but are absent from static pre-fight summaries. A notable limitation of the current approach is the heavy reliance on UFC betting odds, which – while highly informative – are unknown and may incorporate adjustments unrelated to fighter skill and unavailable to the public. As a result, the model's ceiling is partly constrained by the quality and transparency of the betting odds data it uses. In summary, we do not recommend the use of this model to predict UFC fights (J. McDermott, 2025).

References

1. Ultimate UFC Dataset. (n.d.). Retrieved December 9, 2025, from https://www.kaggle.com/datasets/mdabbert/ultimate-ufc-dataset
2. Exploring Patterns and Predictors in UFC Fight Outcomes. (n.d.). Retrieved December 9, 2025, from https://www.stat.cmu.edu/capstoneresearch/fall2024/315files_f24/team11.html#Results
3. Akoglu H. (2018). User's guide to correlation coefficients. Turkish journal of emergency medicine, 18(3), 91–93. https://doi.org/10.1016/j.tjem.2018.08.001
4. (J. McDermott, personal communication, Dec 9, 2025)