Applying Machine Learning Algorithms to Predict UFC Fight Outcomes

McKinley McQuaide
mcquaide@stanford.edu

Abstract

Sports betting is a $155 billion industry. Fighting ranks among the top in the industry, and the Ultimate Fighting Championship (UFC) is currently taking steps to push it even further. Mixed Martial Arts (MMA) fighter statistics involve everything from skill centric values such as wins and significant strikes landed to physiological measurements such as height and reach. There are over one hundred different features up to analyze before any given fight, and machine learning can be used to best understand which are most relevant, and to indent trends and predict the outcomes (win/draw/loss) of each fight. Specifically, this study explores how well Generalized Linear Models, Multilayer Perceptrons, Decision Trees, and Gradient Boosting classifiers can predict the outcomes of future fights based on the all complete fight data in UFC record history, which spans back to 1993.

Introduction

Sports betting is a $155 billion industry. Fighting ranks among the top in the industry, and the Ultimate Fighting Championship (UFC) is currently taking steps to push it even further. The goal of this study is to explore our ability to predict the outcome of UFC fights based on the each match's pre-fight statistics using Generalized Linear Models, Multilayer Perceptrons, Decision Trees, and Gradient Boosting classifiers. An accurate prediction model could both inform the best placed bets (and potential risk associated) for each fight, but also could provide insight to coaches when accepting fights to begin with, simply by looking at the opponents statistics relative to their fighter. It could also be used to help to identify which features are most significant in this prediction.

Related Work

Similar studies have been performed such as *A Comparative Study of Machine Learning Algorithms for Prior Predictions of UFC Fights*[1], which observed the efficacy of machine learning models based on Perceptron, Random Forests, Decision Trees classifier, Stochastic Gradient Descent classifier, Support Vector Machine, and K-Nearest Neighbor in time series. This and many studies observe datasets spanning much smaller timelines, and the resulting accuracies were 55.7%, 58.4%, 51.7%, 55.1%, 57.8%, and 55.7% respectively.

Dataset and Features

There are over one hundred different fighter statistics on UFCStats for each of the 5,144 fights in UFC record history extending from 2019 back to 1993, which include information such as fighters' height, weight, reach, and stance, as well as statistics such as win streaks, strike percentage, guard passes, and strikes landed by location. The dataset used in this study was scraped from UFCStats and statistics pertaining to the circumstances of the fight(i.e.location, number of allocated rounds, etc.)were removed to as they were deemed irrelevant and did not align with the fighter-centric intention of the study. The set was then cleared of samples with incomplete feature sets leaving 3,355 complete samples spanning from 1997 to 2019 with 134 features.

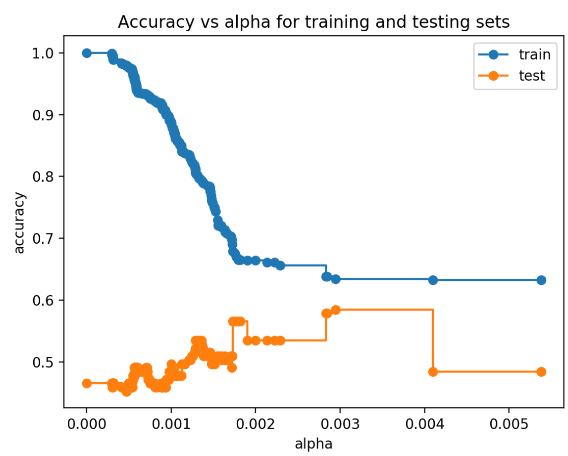
The testing and training sets were split using k-fold validation for time-series model with k=20 folds across the 21 years to determine a more representative accuracy of future predictive ability. Within each fold, the training data inputs were centered and scaled, and the scale was then applied to the test set inputs before fitting.

Method

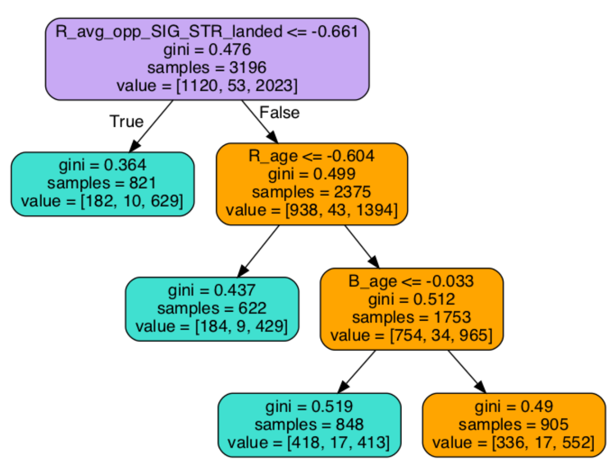
Because there is a discrete (win/draw/loss) outcome, this is a multi class classification problem. Four classification models[2] were used in this study:

1. Generalized Linear Model: Stochastic Gradient Descent was applied with a perceptron loss function.
2. Neural Network: Multilayer Perceptron[3] was used with L-BFGS to optimize the log-loss function. This optimizer was chosen because of its ability to converge faster and preform better on large datasets. While the whole dataset is not small, this is important for the earlier folds, particularly given the higher k value. Here, two layers were used and a grid search was preformed to find the optimal layer size: five neurons in the first layer, and two in the second.
3. A Decision Tree Classifier was applied with minimum cost-complexity pruning with measured effective node alpha as follows[4]:

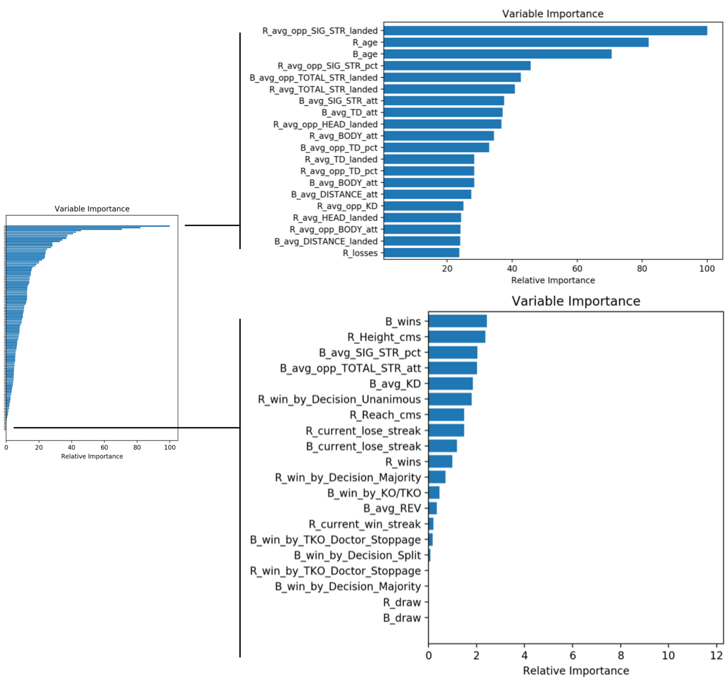
$$R_\alpha(T) = R(T) + \alpha|T| \qquad \alpha_{eff}(t) = \frac{R(t)-R(T_t)}{|T|-1}$$



Where complexity parameter identified at peak accuracy where ccp α = .003. The decision tree is visualized as follows:



4. A Gradient Boosting Classifier was applied with a multinomial deviance loss function and a learning rate of 0.01, producing the following relative variable importances:



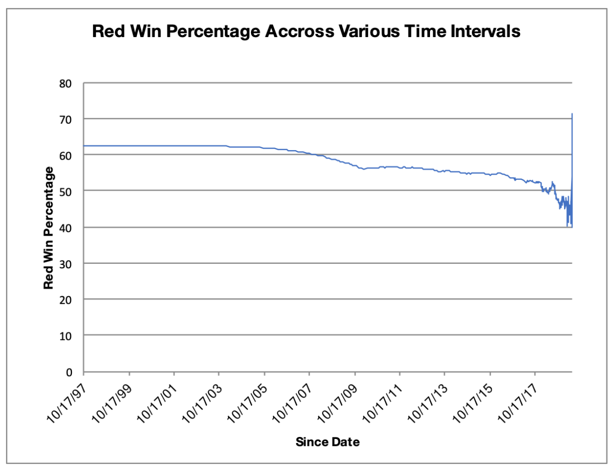
Results and Discussion

After accounting for overfitting and creating the models, the accuracy was computed averaging over the k-folds using Mean Squared Error (1) and the following results were computed:

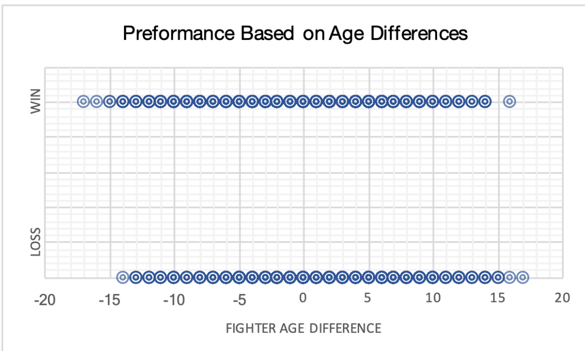
(1) $$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

	Average Training Accuracy (final fold: 3187)	Average Testing Accuracy (k * 168)
Stochastic Gradient Descent	72.876%	58.018%
NN: Multilayer Perceptron (L-BFGS)	89.566%	57.736%
Decision Tree (ccp α = .003)	80.222%	60.252%
Gradient Boosting	88.583%	61.226%

Overall, Gradient Boosting preformed with marginally highest accuracy, however the average test accuracies all land around 60%. The difference between training and testing accuracy is still very high, which can likely be attributed in large part to the even distribution of weights when averaging the k-folds. The earlier folds have much more extreme differences between training and testing sets which is to be expected by the smaller quantity of samples, however it is important to preform a time series k-fold in order to best identify the predictability of the models for future inputs. Observing the variable importances, correlations were explored between simply being the red fighter and winning. While across the entire dataset, the red fighter wins 62.6% of the time, which would make simply guessing red outperform all of these models, looking at the red fighter win percentage over time, as in the graph below, highlights the change in trends over time and supports using incremental training sets via time series k-fold.



Observing this trend indicates that the models would, in fact, preform more accurately in present day.
Another observation from the variable importances relates to fighter age. The age difference from red to blue fighter was then plotted against wins yielding the following distribution:



Here, the age difference represents the age of the red fighter minus the age of the blue fighter. This highlights the subtle suggested relationship between age difference and wins.

Future Work

Moving forward, the parameters should be further adjusted to improve the accuracy of predictions, continuing off of the initial grid search, perhaps replacing it with a randomized search, and applying similar functions in other areas of the study. Additionally, if a pass is taken removing irrelevant parameters, then previously removed samples may be added back in as they may become complete. Repeating this would enable more of the data to be used.

In order to address the high delta between training and testing accuracies, the accuracies of each fold could be redistributed, either exponentially or linearly with the size of the data it was modeled from.

Once the parameters are further refined, and the predictability of the models are fully optimized, the interface of the models needs to be designed in a way that affords a fluid interface for usability each time there is a new fight. Particularly for the models which use a large number of the features, the scraping of fighter stats and inputting to the model should be steam-lined, where it can be scaled and output comprehensive predictions.

References

[1] Hitkul, K. Aggarwal, N. Yadav, and M. Dwivedy, “A Comparative Study of Machine Learning Algorithms for Prior Prediction of UFC Fights,” *Harmony Search and Nature Inspired Optimization Algorithms Advances in Intelligent Systems and Computing*, pp. 67–76, 2018.

[2] *Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.*

[3] *Kingma, Diederik, and Jimmy Ba. “Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980 (2014).*

[4] *L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, Belmont, CA, 1984.*

https://drive.google.com/drive/folders/1v02rr9dGL3ISHIf3rLru6jeW5IxENF6R?usp=sharing