UFC Fight Winner Prediction

Kyle Walsh (kew96), Brady Dickens (brd62)

Abstract:

Since the UFC first began in 1993, statistics have been accumulated over the years documenting each match and fighter. This paper is a summary of various data mining and machine learning methods used to try to gain a competitive edge in predicting the winner of a fight before it happens. By inspecting the individual fighter statistics in previous fights and characteristics, we trained models using basic logistic regression as well as more sophisticated models, such as Random Forests or XGBoost to improve our classification accuracy. With over 140 predictors in our data, it was important to identify which predictors may better contribute to an accurate model, then use these findings to fit new models in an effort to make the best prediction for future UFC fights.

Dataset: UFC Fight Historical Data from 1993-2019

(https://www.kaggle.com/rajeevw/ufcdata?select=preprocessed_data.csv)

Introduction:

Gambling and sports betting has always been a popular event for many people, ambitious to try their luck and make fast money. When thinking about gambling, particularly in sports, winnings and losses are always assumed to balance out, according to the law of large numbers. Beating the odds can be extremely difficult as bookmakers set very enticing odds and they have years of experience in setting these values along with taking in the general public's opinion and changing the odds accordingly. However, bookmakers can, at times, place the wrong odds on certain events before they can correct their mistakes using the public's opinion or someone can even make a bet with a friend as they are about to watch a sporting event. To "beat" these odds, a person must choose the winner greater than half of the time, essentially making them pick winners more often than a simple coin flip would.

Our team has achieved testing accuracies on our best models that slightly outperform these random coin flips. Overall, this is a very challenging task to predict winners of fights because in-fight statistics, training styles, and general human knowledge are very difficult to incorporate into a model. By training multiple models and achieving very similar errors, we believe that there is little more to be done using our current set of data as it is. To improve our model further, we would need more data, along with more, sophisticated metrics and more time to explore different variable interactions to test how they affect our models.

Initial Data Preparations:

The initial dataset is composed of 5145 individual fights, with 145 predictors from location and fighter names to specific fight stats like average number of punches landed. The bare dataset contained many rows of missing data from fighters that would disrupt our models, but the dataset also provided a preprocessed dataset that simply removed all rows with missing fighter information. Instead of preprocessing the original data ourselves, we used this preprocessed form that still contained 3592 observations to base our models off of, which generalized the fight winner as either Red, Blue, or Draw. After creating dummy variables for the target variable, "Winner," we discovered that Red fighters won 2380 of the fights, which is about two thirds of the fights. In an effort to balance the outcome numbers, we experimented with duplicating all Blue winner rows. This resulted in more flexible, skewed models that performed poorly with logistic regression, while overfitting using KNN despite high training accuracy. Unfortunately, we were not able to correct this bias and thus we changed our expectations to be able to predict over two thirds of the fights correctly which we would be able to achieve if we solely predicted that the red fighter wins every fight.

When initially experimenting with various models, we attempted to train on the original preprocessed training set. After training a basic logistic regression model on this training set, we discovered a slightly greater than two thirds accuracy but wanted to work with more indicative data and predictors. We created a correlation matrix to gain some insight into what fight metrics may be related or lead to better findings. The matrix created was too large to draw conclusions

from, but we examined the top 25 predictors correlated with the “Winner” column to see what fighter statistics could be most relevant and correlated to fight winners.

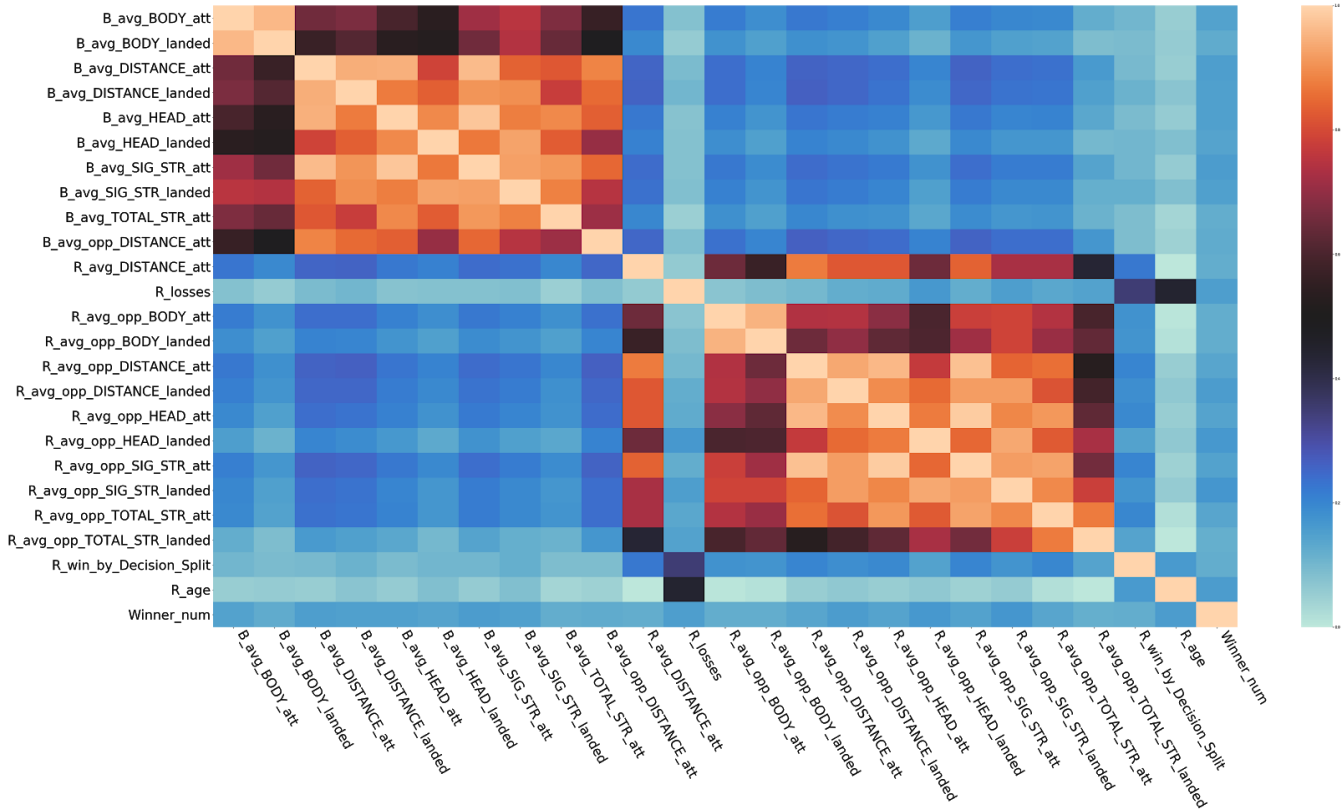

Figure 1: Correlation matrix consisting of the top 25 predictors most correlated with the “Winner” target variable.

Since this revealed that a lot of previous average fight metrics were important, we created a new dataset of interaction terms between Red and Blue fight metrics. For example, we took a statistic like “Red Average Body Shots Attempted” and divided it by the “Blue Average Body Shots Attempted” predictor to create a ratio of the fighters instead of distinct columns. This consisted of a Laplace Smoothing parameter as well to account for incomplete data or zero values. These ratio columns were then added to the more general predictors like height, weight, and record of the fighters to reduce the training data size from 159 predictors down to 112. Aside

from being more accurate than raw averages, this interaction term dataset will help with general time complexity and memory of training future models.

We then split the dataset into train, validate, and test sets to hopefully create the most accurate models possible. The dataset was broken down into 60% training, 20% validation, and 20% test sets.

Prediction Models:

After solidifying our separate datasets, we trained many models including ones using logistic regression with L1 and L2 regularization, KNN, and tree based models. These were all trained on our training set created from the interaction terms implementing smoothing parameters and utilizing cross validation to select the best parameters, and attempt to estimate our testing accuracy. Throughout the training process, accuracy was our primary decision metric, followed by AUC if the accuracies were too similar to recognize a significant improvement from one model to another. We focused on accuracy because that is the ultimate objective, picking as many correct winners as possible.

Our initial logistic regression model gave us 67.3% training accuracy, which is just better than the two thirds threshold of always guessing Red. This solidified our thoughts of having a baseline score of approximately 66% and now moving that value up to 67% for the rest of our models. L1 regularization models were created for various regularization parameters from 0.1 to 1 using cross validation in which we achieved a range of 64.8% to 73.9% accuracy. We trained L2 Ridge Regularization models in the same way to obtain just slightly less accurate models by about 1%.

KNN models were trained to determine which value of k would achieve the best cross validation scores. We trained models with k values from 1 to 100 and observed the following trend as k increased in Figure 2.

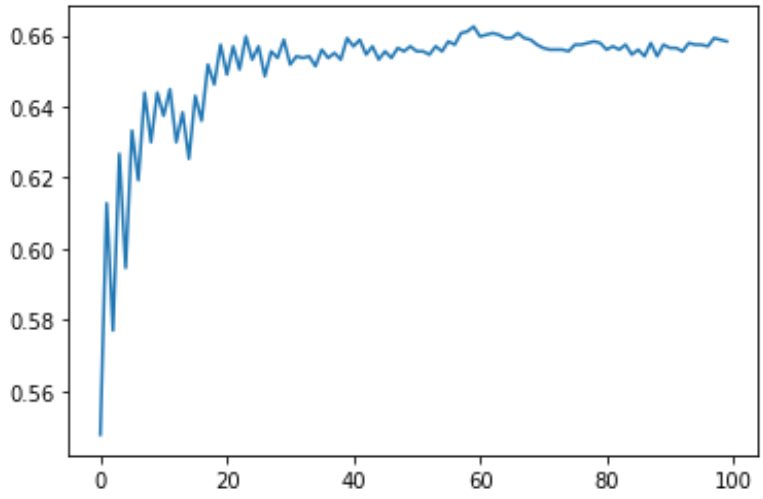

Figure 2: kNN accuracy value plot vs k values (1-100)

The maximum accuracy was achieved when k = 59, but still with only 66.2% accuracy. We concluded that kNN may not have been a great classification model because of the large number of predictors and unpredictable nature of fight results once the match begins.

Lastly, we trained four different tree-based models to see if we could find even more accurate classifiers. The first were basic decision trees with maximum number of features from 1 to 100, resulting in a maximum accuracy of 60.8%. Next, we trained 20 bagging models with values from 1 to 20 of the number of estimators, to achieve maximum training accuracy of 65.5%. The final models were Random Forests and XGBoost trees that gave us greater accuracy of 67.4% and 67.7% respectively.

After training all of our models, some performed significantly worse than expected with a decision tree achieving 60.8% cross validation accuracy. We narrowed our ideal list down to three techniques, consisting of a lasso regularization with basic logistic regression model, a

random forest tree model, and a boosting tree model implementing XGBoost[1]. We decided to only include lasso regularization and not ridge regression, which achieved 73.35% cross validation accuracy, because of the added benefit from lasso of forcing some coefficients to zero, helping with model interpretability.

After the regularized logistic regression models, random forests and XGBoost had the next best cross validation accuracy with XGBoost slightly beating random forests both in cross validation accuracy and AUC scores. When choosing the correct parameters for the random forest model, we iterated through many combinations of parameters and obtained our best cross validation error with 150 trees, splitting based on entropy, 5 samples at minimum to split, and no pruning[2].

Similar to our random forest model, we started with the default inputs for our XGBoost model and slowly tuned the parameters using cross validation accuracy to choose the best parameters at each iteration. We first focused on tuning the maximum depth of each tree and the minimum child weight (the minimum number of samples required to split a node). After this, we tuned the parameter gamma, which represents the minimum loss reduction needed to split on a node. Our next focus for tuning were the parameters referring to the ratio of points to use to train each individual tree from the total set of options. Lastly, we tuned the regularization parameters for the weights of the splits on each predictor.

Overall, based on our cross validation accuracies and AUC values, we would have to choose lasso regularization with logistic regression as our top model. Unfortunately, all of our

[1] Parameters listed in index for these chosen models.
[2] Full lists of iterated parameters shown in appendix.

chosen models performed very similarly on testing data and were marginally better than our goal of an accuracy above two thirds.

Model	Cross Validation Accuracy	AUC	Testing Accuracy
L1/Lasso Regularization	73.87%	0.5572	67.87%
Random Forest	67.36%	0.5371	67.59%
XGBoost	67.66%	0.5641	68.57%

Conclusions:

While lasso regularization on a basic logistic regression model showed a lot of promise with its cross validation accuracy, the testing error indicated the complexity of accurately predicting the winner of a UFC fight solely from pre-fight statistics. Ultimately, more predictors and more data are required to train more robust models. To continue improving our models, after obtaining more data and predictors, we would then train models specific to each weight class and gender. UFC fighters can have extremely different fighting styles as fighters increase in weight and across genders, heavily influencing the important statistics for each fight. As the number of predictors increases, we would also start to explore principal component analysis more, hopefully further improving models and creating better visualizations of the data.

Appendix

Random Forest Iterated Parameters:

- Number of Trees: 50, 67, 83, 100, 117, 133, 150, 167, 183, 200
- Splitting Criteria: gini index, entropy
- Minimum Samples to Split: 2, 3, 4, 5, 6
- Pruning Parameter: 0, 0.25, 0.5, 0.75, 1

Lasso Regularization Final Parameters:

- Tolerance: 0.0001
- Alpha: 0.4

Random Forest Final Parameters:

- Number of Estimators: 150
- Splitting Criteria: entropy
- Minimum Samples to Split: 5
- Minimum Samples per Leaf Node: 1
- Maximum Features to Consider per Tree: $\sqrt{\text{total number of features}}$
- Alpha: 0

XGBoost:

- Learning Rate: 0.01
- Number of Estimators: 50,000
- Maximum Depth: 8
- Minimum Child Weight: 4
- Gamma: 0.1
- Subsample: 0.6
- Column Sample by Tree: 0.7
- Objective: binary:logistic
- Alpha: 0.01
- Lambda: 1