

CS 221 Final Report

Team: Luis Arizmendi, Austin Salcedo, Saba Weatherspoon, Matthew Torre

[Colab](#)

[Link to Slides Used in the Video](#)

Introduction

Many Mixed Martial Arts (MMA) fans like to speculate on who they think will win a fight, and many fans will place bets on who they believe will win or who will perform better/worse. The problem is that there are countless factors that go into this speculation, such as each fighter's: fighting style/fighting background, power level, natural "walking" weight, speed, endurance, striking skills, grappling skills, experience level, the event venue, the referee, etc.

Considering the utility of informed decision making from a multitude of factors, the task at hand is to create a neural net classifier that can take as input all of the statistics of two fighters, and output an accurate prediction of who would win a fight between the two.

Comprehensive Literature Review on Predicting MMA Fight Outcomes

Introduction: Mixed Martial Arts (MMA) fans and bettors often speculate on the outcomes of fights, influenced by numerous factors such as fighting style, physical attributes, experience, etc. Accurate predictions can enhance decision-making and betting strategies. The surrounding literature we looked at is poised at predicting MMA fight outcomes, focusing on the use of logistic regression and other machine learning models.

Key Findings:

The multilayer neural network model showed a more than 10% increase in accuracy over previous models. The use of comprehensive features and advanced evaluation metrics provided a robust framework for MMA fight prediction.

Logistic Regression and Random Forests:

Prasetio et al. (2016) applied logistic regression to predict football match outcomes, achieving an accuracy of 69.5%. The model used significant variables such as home offense, home defense, away offense, and away defense. Logistic regression, as a linear model, provides interpretability through regression coefficients, which highlights the relationship between input features and the outcome.

Random forests, as used in various sports prediction studies, combine multiple decision trees to improve prediction accuracy and reduce overfitting. These models are robust to noise and can handle a large number of features, making them suitable for complex prediction tasks like MMA fights.

Bayesian Networks:

Bayesian networks model probabilistic relationships between fight variables, providing insights into the dependencies and interactions that influence fight outcomes. These models are useful for incorporating prior knowledge and handling missing data.

Comparison with Neural Networks:

Bayesian networks offer a probabilistic framework that can be integrated with neural networks to enhance prediction capabilities. Incorporating behavioral factors into neural network models can improve their accuracy and reliability in predicting MMA fight outcomes. Sharma's paper building out a Multilayer Neural Network Model for MMA fight outcomes, is a study that concludes that a convolutional neural network model can effectively forecast binary class outcomes in MMA with a 10% accuracy improvement over previous models (G. Sharma and A. K. Uttam, 2021). Future research could explore various data balancing techniques, different neural network architectures, and enhanced feature engineering to further improve performance. In addition, applying these methodologies to other combat sports could help validate and extend the model's applicability (G. Sharma and A. K. Uttam, 2021).

Comparison with Hitkul et al. (2019)

The paper by Hitkul et al. presents a comparative analysis of different machine learning algorithms to predict UFC fight outcomes. Their approach is a typical example of applying traditional machine learning methods like logistic regression and random forest to historical data, fighter records, and physical attributes. They emphasize the importance of skill and physical attributes in making predictions, especially when the skill levels of the fighters are similar.

Conclusion

The prediction of MMA fight outcomes has evolved from simple linear models like logistic regression to sophisticated neural networks. Multilayer neural networks, with their ability to capture complex interactions and handle high-dimensional data, have actually shown significant promise in this domain. However, given the robustness and interpretability of logistic regression, it remains a viable method for this task. Logistic regression models are very capable at effectively highlighting the relationship between input features and outcomes through regression coefficients, leading to valuable insights into the factors that influence fight results. The prediction of MMA fight outcomes has used everything from simple linear models like logistic regression to more sophisticated neural networks and ensemble methods.

Dataset

The dataset is a csv file from [Kaggle](#) (the same dataset used in our main [reference](#)'s paper) containing information about different UFC fights including most notably the fighter names, date, winner, win by unanimous decisions, win by knockout, number of wins, number of losses, ect. The data is made up of 4896 fights with a combined 119 attributes including fight odds, physical attributes (height, weight, etc.), fighters' records and rankings, all compiled to a csv file which we parse through using the pandas package to ultimately turn into a feature vector in which each feature is its numerical value, and class which was 1 meaning the red side fighter won and 0 if the red side fighter did not win; this also means that draws count as the red fighter not winning.

Data Cleaning

In order to closely compare our results with our reference paper, we decided to use the same feature engineering techniques that our reference used in their paper. This consisted of removing unnecessary predictors, combining like predictors, and filling n/a values with the appropriate actions. For example, we decided to remove certain data like location, referee and dates are unlikely to have a significant impact on which fighter would win. Any booleans were turned into ones and zeros and non-numerical values were converted to numerical values by using a label encoder in order to use the data as a feature vector.

Baseline

As we develop further we use a baseline which will choose the fighter with a higher win rate with probability p , p is changeable. Currently we have a $p=1$, so in theory it will perform the best at this point. In reality it performs with a loss of about 0.4 typically. (Loss = $(1/\text{Number of fights})(\text{Sum of: prediction})$ in which a prediction = 1 if wrong and 0 if right. So we are correct ~58% (Accuracy) of the time with this baseline. We have another baseline, baseline - 2, which will choose the correct winner with probability p and collects the loss the same way. Ideally our solution and baseline perform somewhere in between $p = 0.5$ and $p = 1$ for baseline - 2, in terms of accuracy.

Since our model uses cross-entropy loss which accounts for values not a right or wrong but with probabilities, which we feel is a better representation of our models performance, we will rely on the accuracy of our baseline rather than the loss for comparing performances, as the losses are calculated differently. In this case we define accuracy as a percentage of correct predictions vs all predictions.

Main approach

We largely decided to model our approach after our [first reference listed in our proposal](#), which found that logistic regression had the highest accuracy amongst many other ML implementations in predicting the winner of a fight based on the fighters' stats. Here, we attempted to replicate the results of their paper and then explore what alterations might improve the performance of a logistic regression implementation, as well as explore how the implementation of neural networks on the same problem might compare.

First, we used Pandas to read in our CSV data and clean the data as mentioned above in the **Dataset** section. Afterwards we followed some tutorials (Loeber and Coelho), referencing them to make a logistic regression model, using their calls for pytorch to put into our model. The next paragraphs outline the references and what they did, and how and why we decided to follow them.

To implement this model we first divided our feature vector and labels into testing and training sets, normalized via `sc.transform` our vectors and made tensors so that we could use pyTorch.

Next we made the Logistic Regression model with neural network layers, however we only have 1 linear layer. As a forward pass, we use the sigmoid function because we are trying to do a binary classification (whether red won or not). Another notable feature is we are using a learning

rate of 0.1, cross entropy loss function for loss, and are moving through are data with stochastic gradient descent. Other than the learning rate, most of this is derived from our references.

To train our model we chose a high number of epochs, 4000, and used our reference to use functions that made this simple. The strategy in this case is to predict a class from our model, compute the loss using entropy loss, do a backward pass, update our weights via a stochastic gradient descent run. Afterwards we computed our accuracies which is discussed in the *Evaluation Metric* section.

To further our investigation, we decided to implement some feed-forward neural networks (NN) on the data as well. Our first one was a simple NN with just two hidden layers (first layer with 128 neurons and second layer with 64) which we used as a baseline NN. In the second “improved” NN (a modified version of the first), we implemented dropout layers, (a regularization technique to reduce the chances of overfitting the data) after each hidden layer. Both NNs used ReLU activations and a sigmoid output.

Evaluation Metric

Quantitatively, similar to our [first reference listed in our proposal](#), we will calculate an accuracy score using logistic regression and compare it to the accuracy using both feed forward neural networks and feed forward neural networks with dropout loafers.

Accuracy is defined in terms of the percentage of times our model was correct compared to the actual outcome of a fighting match in the data.

Results & Analysis

METHOD	ACCURACY
Logistic Regression	66.4%
Feed Forward Neural Net	63.4%
Feed Forward Neural Net w/ dropout layers (regularization technique)	65.2%

Results: The reference paper linked above obtained an accuracy of 66.7% using Logistic Regression with their baseline dataset, whereas our main approach (logistic regression implementation) obtained an accuracy of 63.4%. Our precision was 71.4%. (Please refer to the Baseline section for baseline results). For our first feed-forward NN (no regularization), we obtained an accuracy of 63.4%. For our second feed-forward NN (with regularization), we obtained an accuracy of 65.2%.

Analysis: We are happy with our results, but we are still skeptical about potentially overfitting the data with our logistic regression approach. A big change from our progress report that we submitted earlier in the quarter is that, in this current approach, we transitioned to using the same exact data and feature engineering techniques as our main reference paper did. This way, we were able to closely replicate their results (although, only a few tenths of a percent weaker in accuracy). Since we felt comfortable with the LR replication, we then attempted to reach better results with our NNs, yet to no avail. We speculate there could be many reasons for this, such as maybe we need a larger dataset for the neural network to perform better since they work better on more complex datasets. We also suspect more extensive work could be experimented with in the area of feature engineering and normalization/scaling, yet time did not allow for us to test these theories out.

Error Analysis

We conducted several experiments to analyze the properties of our system, both its strengths and weaknesses. We ran all of our models and calculated the accuracy by comparing the output of the model to the actual outcome of a fight as described in our dataset. We were surprised by the fact that our logistic regression model still worked better than our feed forward neural network model.

To detect overfitting, we evaluated the model's performance on training versus validation data, which revealed some potential overfitting as indicated by lower training loss compared to validation loss. This prompted us to evaluate the need for regularization techniques and cross-validation. Finally, comparing logistic regression with feed-forward neural networks showed that logistic regression performed slightly better without regularization but was close to the regularized neural network. This showed us that logistic regression's effectiveness in the presence of well-balanced and relevant features. The experiments revealed that proper data balancing, feature selection, and regularization are pivotal for improving model performance.

Future Work

Moving forward, we could customize our model further by testing other types of neural networks. In our project, feedforward neural networks were actually slightly less accurate than our model using logistic regression. The reason behind feedforward neural networks being slightly less accurate in our dataset could be a point of further study, as well as finding another neural network model that would significantly increase the prediction accuracy.

We could also look to combine more datasets to build a larger one from other sources and experiment with more feature engineering, e.g. we could evaluate referee data in each match to see if fairness plays a significant role in the outcome of UFC fights.

References - Please include a reference section with properly formatted citations. (Not included in 4 page limit).

OpenAI. (2024). ChatGPT (Jun 3 version) [Large language model]. <https://chat.openai.com/chat>

Hitkul, Aggarwal, K., Yadav, N., & Dwivedy, M. (2019). A comparative study of machine learning algorithms for prior prediction of UFC fights. In Harmony Search and Nature Inspired

Optimization Algorithms: Theory and Applications, ICHSA 2018 (pp. 67-76). Springer Singapore.

James, L. P., Robertson, S., Haff, G. G., Beckman, E. M., & Kelly, V. G. (2017). Identifying the performance characteristics of a winning outcome in elite mixed martial arts competition. *Journal of science and medicine in sport*, 20(3), 296-301.

Loeber Patrick. PyTorch Tutorial 06- Training Pipeline: Model, Loss, and Optimizer (2020). https://www.youtube.com/watch?v=VVDHU_TWwUg&list=PLqnsIRFeH2UrcDBWF5mfPGpqQDSta6VK4&index=7

Coelho Carlos. Logistic Regression - PyTorch. <https://medium.com/@carlosrodrigo.coelho/logistic-regression-pytorch-956f96b28010>

Prasetio, D. (2016, August). Predicting football match results with logistic regression. In 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA) (pp. 1-5). IEEE.

Sharma, G., & Uttam, A. K. (2021, October). Multilayer neural network model for mixed martial arts winner prediction. In 2021 5th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1-5). IEEE.

Vu, B. (2022). Ultimate Fighting Championship's Fights Analysis. ISYE 7406.