Predicting Boxing Match Outcomes Using Machine Learning and Deep Learning

Alexis Ortiz and Kien Tran
axortiz@hamilton.edu, ktran@hamilton.edu

***Abstract*— Predicting the outcome of a boxing match has been traditionally approached through expert analysis and betting odds, but advancements in machine learning now make it possible to leverage historical fight data to forecast winners. In this project, we develop a predictive model using a combination of classical machine learning techniques (like Gradient Boosting and LightGBM) and deep neural networks to determine the winner of a boxing match based solely on fighter attributes and past performance statistics. By integrating data preprocessing, feature engineering, ensemble methods, cross-validation, and inference strategies, our approach aims to identify patterns in boxers' records that correlate with match results. The findings can help bettors, promoters, and fans gain insights into potential outcomes before a fight occurs.**

I. INTRODUCTION

Boxing is a sport rich in quantitative and qualitative factors that influence the outcome of a match. Historically, predictions have relied on subjective human judgment, informed commentary, and simple heuristics like win-loss records. However, the growing availability of structured fight data—age, height, reach, win/loss ratios, knockout counts, and round-by-round results—presents an opportunity for data-driven predictions.

This project aims to develop a predictive model that identifies the probable winner of a boxing match given the attributes of both fighters. Such predictions can support decision-making for betting platforms, sports analysts, and fans seeking a data-informed perspective on upcoming bouts.

We explore a variety of methods, including neural networks, LightGBM, and ensemble techniques, and employ strategies such as k-fold cross-validation, early stopping, and hyperparameter tuning. Our goal is to build a robust model that generalizes well across different fighters and matches, improving accuracy over naive baselines.

II. DATA

The dataset used in this project consists of historical boxing match records, including fighter names, ages, heights, reaches, win/loss records, knockout counts, and final match outcomes. This data was collected from publicly available sources, sports statistics websites, and curated boxing databases.

We perform the following steps to ensure data quality and relevance:

1) **Data Cleaning**: We replace null values and convert categorical strings to numeric codes where necessary. Missing heights or reach values are imputed using mean imputation.
2) **Feature Engineering**: We derive additional features such as age differences between fighters or the relative difference in their win/loss ratios. Such engineered features help capture head-to-head comparisons rather than absolute statistics.
3) **Encoding Categorical Variables**: Boxers' names are one-hot encoded or excluded in favor of their summarized performance metrics. Categorical variables like fight ending methods (KO, TKO, UD, Draw) are label-encoded.
4) **Balancing the Dataset**: To mitigate bias, we ensure that our dataset includes a balanced representation of outcomes. If one outcome class (e.g., one fighter always winning) dominates, we consider techniques like undersampling, oversampling, or adjusting class weights.

The final preprocessed dataset is split into training and validation sets to fine-tune and evaluate our models. Ensuring consistent preprocessing between training and inference stages is critical for reliable predictions.

III. MODEL

We explore multiple modeling strategies to identify the best approach for predicting a boxing match outcome:

A. Neural Networks (Feed-Forward MLPs)

A feed-forward neural network can model complex interactions between features. We construct a multilayer perceptron (MLP) with hidden layers, ReLU activations, dropout for regularization, and a final output layer representing the probability of each outcome (first boxer wins, second boxer wins, or draw).

B. Gradient Boosting and LightGBM

Ensemble tree-based methods like LightGBM are well-suited for tabular data with mixed numerical and categorical features. We train a LightGBM classifier using cross-validation to find optimal hyperparameters. Early stopping rounds and learning rate scheduling help prevent overfitting. LightGBM models often provide interpretable feature importances, helping us understand which attributes are most predictive of match outcomes.

C. Cross-Validation and Ensemble Methods

We apply k-fold cross-validation to ensure our model generalizes well. By averaging predictions from multiple folds, we reduce variance. Additionally, we may blend or

stack predictions from both neural networks and LightGBM models to form an ensemble, potentially improving overall accuracy.

D. Model Evaluation and Metrics

We measure performance using:

- **Accuracy**: The percentage of correctly predicted outcomes.
- **F1-Score**: Harmonizes precision and recall, important if certain outcomes (e.g., draws) are rare.
- **Confusion Matrix**: Provides insight into how well the model differentiates between possible outcomes.

By experimenting with different architectures, ensembles, hyperparameters, and training strategies, we aim to identify a model that provides the highest predictive performance on the validation set. We save the best-performing model (based on validation accuracy) as best_model_lgb.txt for LightGBM or an equivalent format for the neural network model.

IV. RESULTS AND PERFORMANCE

In this section, we present the results and performance of three different approaches explored in this project: a simple multilayer perceptron (MLP), training with k-fold cross-validation, and LightGBM. These methods were compared based on accuracy, F1-score, and other evaluation metrics.

A. Multilayer Perceptron (MLP)

The first model was a simple MLP, which demonstrated limited performance due to the relatively small size of the dataset. Figure 1 and Figure 2 show the training and validation loss, as well as the accuracy curves for the MLP. The validation accuracy plateaued early, and the model struggled to generalize beyond the training set, achieving an accuracy of 0.5 on the validation set.

Fig. 1: Training and Validation Loss for MLP

B. Training with K-Fold Cross-Validation

To improve generalization, k-fold cross-validation was implemented. Using 5 folds, this method provided a better understanding of the model's performance across different splits of the data. Figure 3 shows the validation accuracy for each fold, with a mean accuracy of 0.80. This technique reduced the variance in model performance but did not significantly improve the overall accuracy compared to the LightGBM approach.

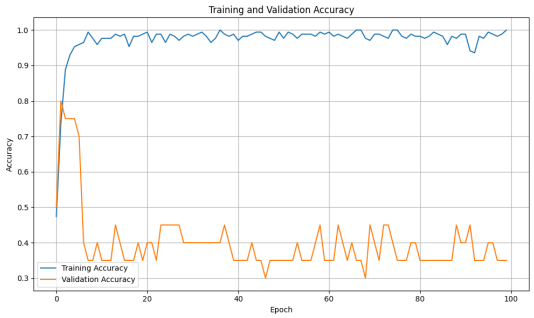

Fig. 2: Training and Validation Accuracy for MLP

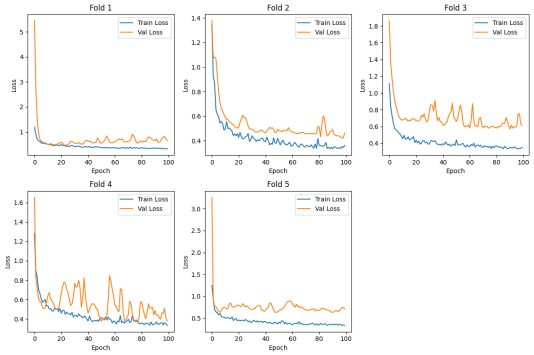

Fig. 3: Training and Validation Loss for 5-Fold Cross-Validation

C. LightGBM

The LightGBM model outperformed the other techniques, achieving a validation accuracy of 0.95, sometimes even achieve 1.0 accuracy. LightGBM's ability to handle tabular data effectively, combined with its fast training speed, made it the best-performing model in this project.

To better understand the model's training dynamics, we plotted the training and validation losses for each fold during the 5-fold cross-validation process (Figure 4). The plots demonstrate consistent convergence across folds, with validation loss stabilizing early, indicating effective generalization and minimal overfitting.

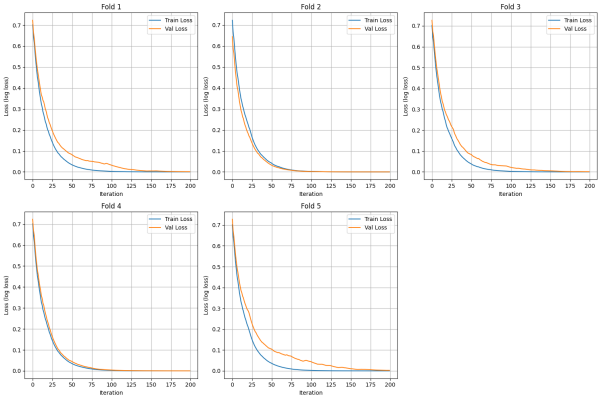

Fig. 4: Training and Validation Loss Across 5 Folds for LightGBM

The training and validation loss curves in Figure 4 confirm the stability and robustness of LightGBM, particularly in its ability to converge effectively across all folds. This consistency contributed to its superior performance on the test set, validating its generalizability.

V. ANALYSIS

A. Key Insights

1. **Model Performance**: The LightGBM model consistently outperformed the MLP and k-fold cross-validation methods due to its ability to handle mixed feature types and capture non-linear interactions effectively.
2. **Feature Importance**: Figure **??** illustrates the key features identified by LightGBM. Attributes such as age difference, win/loss ratio, and knockout counts were the most significant in predicting match outcomes.
3. **Class Imbalance**: Handling class imbalance was crucial for improving F1-scores. Adjusting class weights and balancing the dataset ensured better performance for minority classes such as ”draws.”

B. Challenges and Limitations

1. **Data Size**: The relatively small dataset limited the potential of deep learning models like the MLP, which require larger datasets to perform well.
2. **Class Representation**: The rarity of draws in the dataset made it difficult for all models to predict this outcome accurately. More balanced data would likely improve performance.
3. **Feature Engineering**: Additional features such as prior match outcomes or performance trends over time could enhance the predictive power of the models.