UFC Fight Forecaster

Bryant Ndongmo, Paterne Byiringiro, Gerson Asifiwe

Introduction

The UFC Fight Forecaster project was born out of the necessity to engage fans with UFC events in a more objective and inclusive manner. Leveraging data analytics, our project offers unbiased predictions ensuring every fighter receives equal visibility. By analyzing fighter profiles and predicting fight outcomes, the UFC Fight Forecaster provides insights into potential fight scenarios and showcases similar past fights. This paper presents a high-level overview of our solution, including the problem statement, our approach, data sources, system design, and evaluation highlights.

Our solution employs a comprehensive approach to predict UFC fight outcomes by leveraging various machine learning classifiers and ensemble techniques. Key features such as fighter age, Elo rating, and fight statistics play significant roles in predicting fight results. By utilizing a combination of tree-based, linear, and boosting models, along with ensemble methods, our solution accounts for different patterns and nuances in UFC fights, providing a robust framework for fight outcome prediction.

Evaluation of our model indicates promising results, with the model predicting only the winner achieving an accuracy of 60.96% and an average log loss of 0.656. However, there is room for improvement, and future work will focus on refining the model's performance through more thorough evaluation and potentially incorporating additional features and data sources.

Background

The UFC Fight Forecaster project addresses the challenge of keeping fans engaged with UFC events amid a constantly changing roster, providing a more objective and inclusive way for fans to engage with UFC events. Traditional media coverage tends to focus on high-profile fighters, leaving fans without a reliable, data-driven tool to objectively evaluate all matchups. The project leverages data analytics to provide unbiased predictions ensuring that every fighter receives equal visibility. By analyzing fighter profiles and predicting fight outcomes, the UFC Fight Forecaster offers insights into how fights might unfold and showcases similar past fights, helping fans discover hidden gems and appreciate the full spectrum of talent within the UFC. Ultimately, the project aims to enhance the UFC fan experience by providing a richer, more informed viewing experience, moving beyond media hype, and fostering a deeper appreciation for the sport.

Data

The dataset used for this project was sourced from the 2023 UFC MMA Dataset from Kaggle and includes comprehensive statistics from UFC fights spanning from 1994 to 2023. This dataset is unique due to its normalized structure, incorporating primary and secondary keys for each fight, event, fighter, and fight stat, which enhances data integrity and relational analysis. It contains data on the events, fighters, individual fights, and various key fight statistics, making it highly suitable for fight predictions, exploratory data analysis, and classification models. The primary data collection involved information on fight location, date, fighter characteristics (weight, reach, age, height), match statistics (strikes, takedowns, etc.), and match outcomes (KO, Decision, Submission, etc.). Due to the nature of frequent changes in UFC rules and regulations since 1994 and gaps within data such as the lack of control time for earlier fights, we primarily focussed on fight data from 2014 onwards. For training the model, we utilized this richly detailed Kaggle dataset.

Data summary statistics

	1994 - 2014	**2015 - 2023 pre-processing**	**2015-2023 post-processing**
Total fights	2504	4714	1870
Total fighters	1946	1621	809
Fight outcome stats			
KO	857	1475	615
Decision	985	2330	934
Submission	602	859	321
Other(draws & QD,..)	60	50	0

Scraping

Our data collection process leverages a combination of web scraping techniques and Python programming, employing tools like requests for making HTTP requests, BeautifulSoup for parsing HTML content, and csv for handling CSV files, to gather comprehensive information about UFC events, fights, and fight statistics from UFCstats.com. The process begins with the get_urls.py script, which scrapes URLs of both upcoming and completed events and fights, enabling us to access the necessary data. Subsequently, the events.py and fights.py scripts are invoked to extract details such as event names, dates, locations, fighter names, outcomes, and more, utilizing the aforementioned libraries. Concurrently, the fightstats.py script focuses on retrieving detailed statistics for each fight, including striking, grappling, and control metrics. Once the data is collected, the normalize.py script plays a crucial role in normalizing and structuring the data, adding primary and foreign keys where necessary, and saving the transformed datasets into CSV files for further analysis. Finally, the main script, get_data.py, orchestrates the entire process, executing the scraping tasks based on the specified type of fights (upcoming or completed). This streamlined approach ensures the systematic retrieval and organization of UFC data, facilitating insights and analysis for various stakeholders. This combination of historical data from Kaggle and real-time data scraping ensured a robust and up-to-date dataset, enabling accurate and timely predictions for UFC events.

Motivation

Our solution aimed to employ a comprehensive approach to predict UFC fight outcomes by leveraging various machine learning classifiers and ensemble techniques. Key features such as fighter age, Elo rating, and defensive statistics play significant roles in predicting fight results. By using a combination of tree-based, linear, and boosting models, along with ensemble methods, the solution accounts for different patterns and nuances in UFC fights combining the strengths of individual models, optimizing their predictive power and providing a robust framework for UFC fight outcome prediction.

System Design

The processes outlined below were used to train models to predict winners as well as another model predicting the winner as well as the method of victory.

Feature Engineering

The physical attributes of a fighter have an undeniable influence on their ability as a fighter. Our dataset contained the heights, weights, and reaches for most fighters. However, some of the fighters had missing physical attributes. In order to resolve this, we utilized an IterativeImputer to fill in missing values. We opted for this over using mean or median imputation as every fighter had a weight making it possible to more accurately estimate missing values through iterative imputation taking into account the relationships between different physical attributes, resulting in more precise estimations overall. The next important step was being able to compute the ages of each fighter as well as the time gaps in between their fights. We believed these to be important features to consider as UFC fighters tend to peak at a certain age and long periods of inactivity may be indicative of a fighter declining or struggling with injuries which impact their performance.

We also added columns for each fighter in a fight indicating whether they won or lost and how. This allowed us to see how much success in previous fights predicts future success.
There was some skepticism, however, as this would not take into account quality of wins. This is where Elo comes in. Elo is a rating system that awards points to the winner and deducts the corresponding amount from the loser. The magnitude of this difference is dictated by the Elo

difference before the fight as well as a K-factor that determines how much each fight affects the fighters' ratings. By incorporating Elo ratings, we could not only consider whether a fighter won or lost but also the quality of their opponents and the manner in which they won or lost. Tuning was done to make sure the best K-factor was chosen as well as tuning a multiplicative factor of K for fights won decisively (KO, submission). A very minor time decay of Elo was also incorporated to give more weight to recent performances. This comprehensive approach helped mitigate the limitations of simply counting wins and losses, providing a more robust framework for predicting future outcomes.

Next came the process of actually defining our features. For each fight we had one row for each fighter containing their stats during the fight. We computed running totals of these stats up to the fight as well as their last three fights. From there we were able to compute per fight and per second stats for each fighter over the specified periods. These features accompanied aforementioned features such as elo, age, and days since last fight. In order to predict one fight, we take the metrics of fighter 1 and subtract the stats of fighter 2. In the case of only predicting winners, our target mapped a victory for fighter 1 to 0, and fighter 2 to 1. In order to deal with class imbalance and bias toward predicting fighter 1 or 2 in future predictions, we duplicated the dataset and inverted the features so that the row was fighter 2-fighter 1 instead and switched the targets. This ensured a 50/50 split of wins for either side. Our total number of features was 110.

Model Training

We conducted experiments with various classifiers to determine the most suitable models for predicting UFC fights. We ended up deciding to tune the following models:

Forest Models (Random Forest, Extra Trees):

- **Strengths**: Tree-based models are robust and capable of capturing complex relationships between input features and target variables. They can handle both numerical and categorical data effectively.
- **Weaknesses**: Despite their effectiveness, random forests can be computationally expensive, especially with large datasets.
- **Reason for selection:** boosting models often perform exceptionally well in predictive tasks. We aimed to find the optimal hyperparameters to maximize their performance.

Linear Models (Logistic Regression, Linear Discriminant Analysis):

- **Strengths**: Linear models are simple, interpretable, and often perform well when the relationship between features and target variables is approximately linear.
- **Weaknesses**: They may not capture complex, nonlinear relationships in the data as effectively as tree-based models.
- **Reason for selection**: We decided to tune these models to explore whether a simpler model might perform adequately for our problem, and to ensure that if linear relationships exist in our data, they are captured optimally.

Boosting Models (Histogram Gradient Boosting, XGBoost, LightGBM):

- **Strengths**: Boosting algorithms build strong predictive models by combining the predictions of multiple weak models. They typically perform very well and are highly flexible.
- **Weaknesses**: They are more computationally expensive compared to other models, and may require more tuning to optimize performance.
- **Reason for selection**: Similar to forests, boosting models often perform exceptionally well in predictive tasks. We aimed to find the optimal hyperparameters to maximize their performance.

Tuning these models allowed us to optimize their performance for predicting UFC fights, taking into account both the strengths and weaknesses of each model type. Additionally, comparing their performance against a DummyClassifier provided a baseline for evaluation, ensuring that the selected models outperformed a simplistic, non-learning-based approach.

Our procedure for tuning these models aimed to be as sophisticated as time and resources allowed. Here is the step-by-step process we followed:

1. **Feature Selection and Model Building Function**:
   - We defined a function, `build_model`, to perform feature selection and hyperparameter tuning for each model.
   - This function takes the features, target variable, hyperparameter tuner, and a verbosity flag as input.
   - Initially, all features are considered.
   - Inside a `while` loop, we perform the following steps:
     - Fit the hyperparameter tuner to the features and target.
     - If the model's performance improves, update the best model.
     - Next, we perform iterative feature selection:
       - For each feature:
         - Add or remove the feature from the feature set.
         - Calculate the cross-validated score (5-fold).
         - If the score improves, update the feature set.
       - If the model's performance improves, update the best feature set.
     - Repeat until no further improvement is observed.

For the Extra Trees and Random Forest classifiers, we employed Bayesian optimization (BayesSearchCV) due to the large number of hyperparameters and their sensitivity to tuning. This method allowed us to efficiently search the hyperparameter space. We tuned parameters such as the number of estimators, the criterion for splitting, maximum features to consider at each split, and various parameters governing sample splits and leaf nodes.

Boosting models like Histogram Gradient Boosting, XGBoost, and LightGBM were also tuned using BayesSearchCV. These models have many hyperparameters and are computationally expensive to tune with grid search. We optimized parameters such as learning rate, maximum depth of each tree, subsample ratio of the training instances, minimum sum of instance weight needed in a child, and regularization terms. This approach ensured an efficient exploration of the hyperparameter space.

Linear models like Linear Discriminant Analysis (LDA) and Logistic Regression have fewer hyperparameters compared to tree-based models. Therefore, we used grid search (GridSearchCV) for tuning. For LDA, we optimized parameters such as the solver, shrinkage, and tolerance. For Logistic Regression, we tuned parameters like penalty (L1 or L2), regularization parameter (C), tolerance, and dual formulation.

This comprehensive procedure allowed us to systematically optimize each model's hyperparameters and feature set, maximizing their predictive performance for UFC fight prediction.

Feature Importance

One of the most interesting aspects of this project was determining what features are most predictive when it comes to the outcome of fights. Furthermore it was interesting to see how different classes of models prioritized features not only in the feature selection stage but their significance in the final model.

Linked here are graphs demonstrating the top features for each model for only predicting the winner. Models of the same type produced similar graphs The histogram gradient boosting classifier did not have a feature importance attribute. The analysis of the top features for each model reveals intriguing insights into the factors that significantly influence fight outcomes in UFC matches. Notably, defensive statistics, as indicated by "ag" stats, emerged prominently across various models, suggesting that a fighter's defensive capabilities may be as crucial, if not more, than their offensive skills. Features such as opponent significant strikes accuracy, opponent control time, and reversal rate indicate the importance of defense in avoiding strikes, controlling the fight, and turning the tide against opponents.

Interestingly, some notable features, such as reach and win percentages, were excluded from the top features across all models. This suggests that while reach and win percentages are important factors, other features, particularly those related to defensive capabilities, may have a more significant impact on fight outcomes. Moreover, while striking proficiency was consistently significant, features related to grappling, such as submission attempts and takedown success, also played crucial roles in predicting fight outcomes. These findings highlight the multifaceted nature of UFC fights, where a combination of offensive and defensive skills, as well as a well-rounded skill set encompassing striking and grappling, contribute to a fighter's success in the Octagon. Feature importance for the winner+method model differed as the styles of each fighter became much more important to predict their paths to victory (e.g. submission attempts to predict submission wins)

Ensemble Model

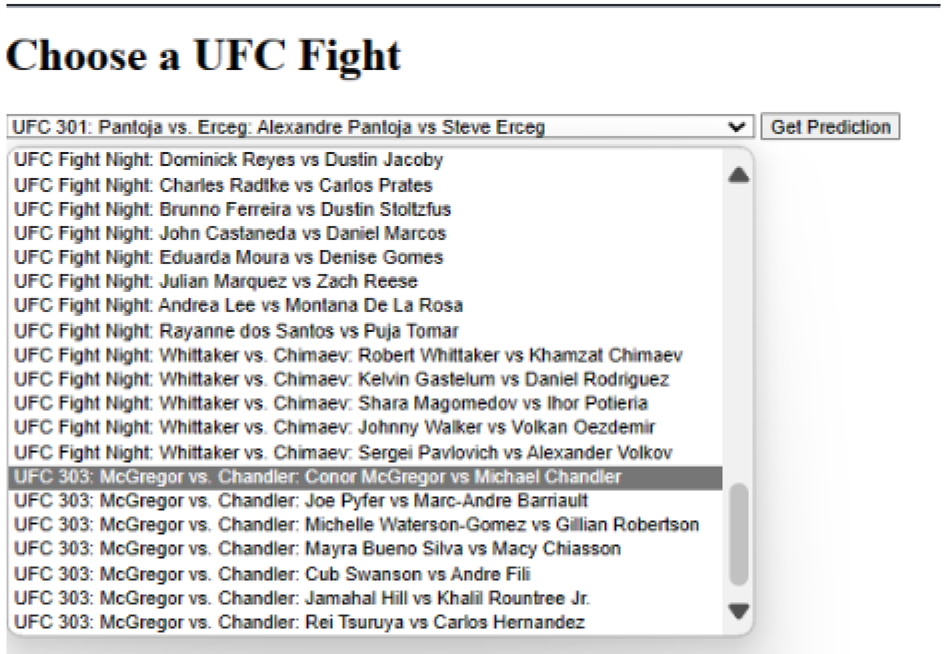
After fitting the models for each classifier, we aimed to leverage their abilities to detect different patterns and investigate whether or not a combination of them can lead to increased performance. Using the same validation folds as the hyperparameter tuning process. The optimal weight for each model was determined to make predictions. This was achieved with a scipy minimize function using the COBYLA method rather than a gradient based method due to the nature of the evaluation metric. By assigning optimal weights to each model, we aimed to create an ensemble that maximizes predictive performance. The weights were determined based on the validation performance of each model, ensuring that models with higher validation scores contributed more to the final prediction. This approach allowed us to leverage the strengths of each individual model and potentially mitigate their weaknesses, leading to a more robust and accurate prediction. This comprehensive approach aimed to exploit the unique patterns detected by each classifier and investigate whether their combination could lead to improved performance in predicting UFC fight outcomes. Below is the optimized set of weights:

Model	Simple Model Weight	Detailed Model Weight
ExtraTreesClassifier	0.02%	1.9%
RandomForestClassifier	0.02%	0.1%
LinearDiscriminantAnalysis	62%	5.3%
HistGradientBoostingClassifier	0.02%	0.06%
LogisticRegression	0.8%	29.9%
XGBClassifier	0.06%	41.5%
LGBMClassifier	37%	21.1%

The results seem to suggest that some combination of linear and boosting models may be best for prediction.

User Interface

The user interface (UI) for the "UFC Fight Forecaster" project is designed to provide an intuitive and engaging experience for UFC fans. The main page displays a dropdown menu populated with all upcoming fights, allowing users to select a specific matchup. Upon selecting a fight, users are presented with images of the fighters along with detailed statistics on how each fighter might win—whether by decision, KO/TKO, or submission. The interface also showcases a list of similar past fights, which helps fans understand what to expect based on historical data. This feature-rich UI is built using HTML templates rendered by Flask, ensuring a seamless and responsive interaction. The use of Bootstrap for styling enhances the visual appeal and usability, making it easier for fans to navigate through predictions and discover exciting matchups. By providing both visual and statistical insights, the UI plays a crucial role in enhancing the overall fan experience, allowing users to make more informed decisions about which fights to watch. Below is a demo of the page and here is a link to the prediction for the highlighted fight:



Evaluation

More thorough work could have been done to evaluate the effectiveness of the final model, but our testing consisted of predicting the outcome of the last 374 fights in the UFC. The model only predicting the winner achieved an accuracy of 60.96% and an average log loss of .656 compared to a validation log loss of 0.628. For reference, according to dratings.com, sportsbooks have achieved an accuracy of 66.4% and an average log loss of 0.613 over the last 1998 fights. Dratings themselves also have publicly available predictions and have achieved an accuracy of 65.4% and an average log loss of 0.622. This suggests that there are several improvements to be made to the model following a perhaps more robust evaluation of its test performance

For our model predicting the method+winner. The model predicted the correct combination with 28.88% accuracy and an average log loss of 1.686 compared to a validation log loss of 1.600. No reference was found to assess this result.

Related Work

Much of the training for this model was inspired by the work done by Dan McInerney on mma-ai.net who himself trained a UFC prediction model using a multi-model ensemble of XGBoost and CatBoost. He boasts a 64.2% accuracy with an average log loss of 0.65. His website provides a guide for anyone who wants to build their own prediction model whether it be for UFC, other sports, or even political outcomes among many others.

Conclusion

By leveraging machine learning classifiers and ensemble techniques, our solution offers unbiased predictions and detailed fight previews.

Although our model achieved promising results, with an accuracy of 60.96% and an average log loss of 0.656 predicting the winner, there are areas for improvement. Future work could focus on refining the model's performance through more thorough evaluation and a much more rigorous feature engineering process. Additionally, the user interface will need to be further developed to provide fans with a more intuitive and engaging experience. Despite its current limitations, the UFC Fight Forecaster project represents an exciting step towards a more data-driven and inclusive approach to UFC event engagement.