Felix Guerrero

Ultimate UFC Dataset Technical Report

1 Knowledge About the Data

The dataset contains 6,528 records with 118 features. Among these features, the datatypes are 60 float64, 43 in64, 14 strings, and 1 bool. Some columns are sparse, for example, all weight classes are treated as separate columns, likely a design-choice, but fighters only belong to one weight-class. Additionally, certain granular features should be ignored to avoid overfitting data. Fighters can win with a finish in various ways. A decision victory can be categorized as split, unanimous, and majority. To reduce the feature count and redundancy, the categories should be merged into one (WinByDecision) and it should be applied to the different finishes. The betting markets offer three methods of victory per fighter; decision, submission, and knockout.

2 Pre-processing Steps

A missing data analysis was performed to find the distribution of missing values across the features. As mentioned before, the weight/rank related features are sparse and have columns exceeding 95% missing values. A threshold-based removal was performed to remove features with missing value ratios of 60%.

The WeightClass variable was converted to an ordered categorical variable, and since the UFC has deprecated “Catchweight” class, it will be dropped.

String processing was applied to all object columns to eliminate any spaces that can interfere with encoding; which is performed with pandas *conver_dtypes()*.

A manual feature removal was conducted to remove variables that seem irrelevant or redundant for a predictive model. For example, fighter names, geographic locations, derived features (e.g RedExpectedValue), and contextual features (e.g EmptyArea, Draws, FinishRoundTime).

After preprocessing, roughly 80 features remain. XGBoost will be used for feature importance to see which contributes the most to the "Winner" target. The results helped form the hypothesis "*are betting market odds more accurate predictors of UFC fight outcomes than fighter statistics, physical attributes, and official rankings*?". Thus leaving only 50 features that can be split into four sets {betting_features, fighter_stats, physical_features, all_features (concatenation of the previous 3 sets)}.

3 Results of Model Experiments

Three supervised-learning models were chosen to test the hypothesis if "*betting market odds alone are strong predictors of fight outcomes*". Accordingly, the four sets are aligned with the binary target (y, Red=1/Blue=0), dropping rows without outcomes. A train/test split (80/20) is performed to preserve class proportions. Numeric odds features are median-imputed (robust to skew/outliers) and standardized. Categorical columns are mode-imputed and one-hot encoded.

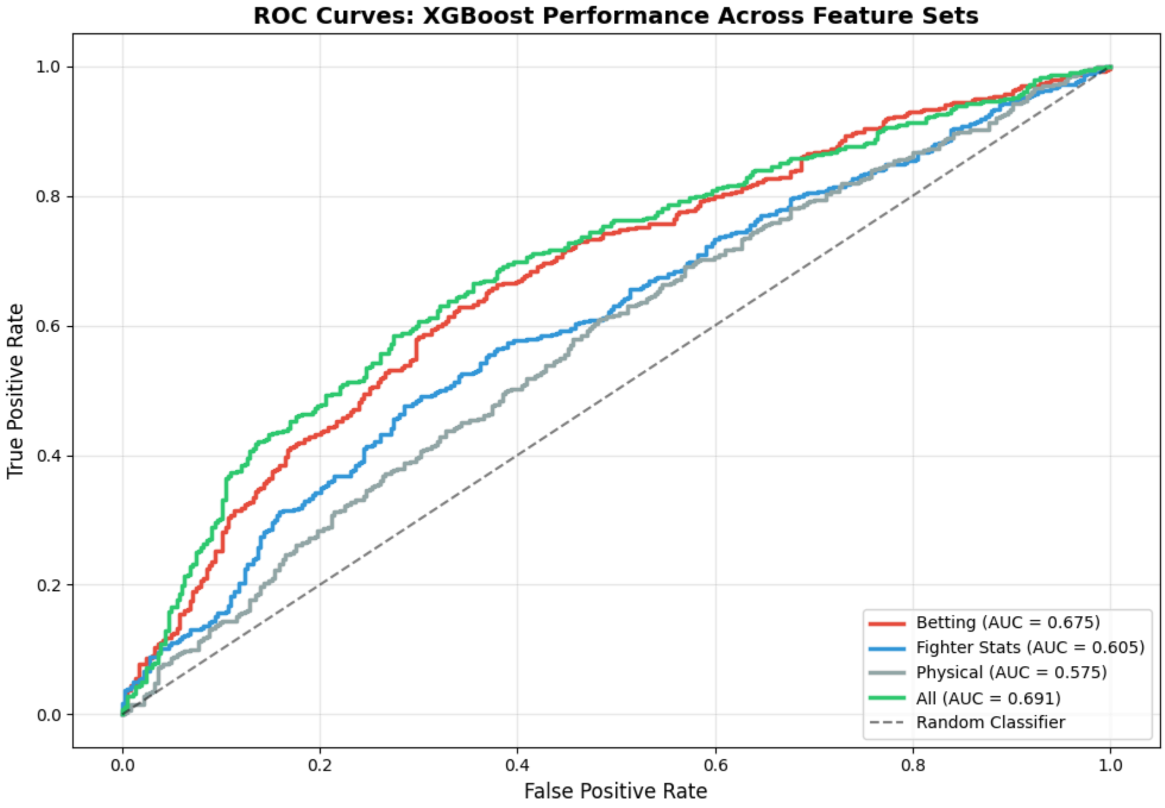
F1, and ROC-AUC are calculated on the three models. Note, decimal numbers are rounded to the nearest hundredths.

Feature set	Model	Accuracy	F1	ROC-AUC
all_features	Random Forest	0.64	0.711	0.70

all_features	Logistic Regression	0.66	0.71	0.70
all_features	XGBoost	0.65	0.71	0.69
betting_features	Random Forest	0.65	0.72	0.70
betting_features	Logistic Regression	0.65	0.71	0.71
betting_features	XGBoost	0.65	0.70	0.67
fighter_stats	Random Forest	0.60	0.72	0.63
fighter_stats	Logistic Regression	0.61	0.71	0.61
fighter_stats	XGBoost	0.59	0.68	0.61
physical_features	Random Forest	0.60	0.72	0.59
physical_features	Logistic Regression	0.57	0.68	0.59
physical_features	XGBoost	0.580046	0.67	0.58

The F1-score represents a harmonic mean of precision and recall. In this case, it shows how well the model performs on a dataset where the target values are possibly imbalanced, finding true positives . Interestingly, most models have similar F1 scores but when predicting with the test set, the physical_features and fighter_stats sets perform poorly against betting_features. This suggests that a high F1 score does not reflect real decision-making accuracy even if it appears that models are correctly identifying winners.

The ROC AUC is a better metric. The ROC (Receiver-operating characterisitc) curve is a visual representation of the true positive rate and false positive rate at every possible threshold. The AUC (area under the curve) represents the probability that the model will rank the positive higher than the negative



Oftentimes, an underdog fighter can have an unexpected victory and cause an imbalance during modeling; which most likely explains the high F1 scores across the feature sets with different models. ROC AUC captures the model's ability to correctly order winners above losers even when probabilities are close. This is important in the UFC analysis if the goal is to estimate how likely each fighter is to win rather than producing an uninterpretable binary decision.

Both metrics show that the betting_features set are the strongest predictors. It performs identical to the set containing all the complex features from extraction. It indicates that betting markets already incorporate relevant information needed to predict fight outcomes. This implies that odds are well-priced and reflect a combination of expert analysis, public sentiment, and historical performance. The market efficiently aggregates information about fighter ability, matchup dynamics, and expected outcomes.

4 Justification for Model Choices

The UFC ultimate dataset works best with supervised learning because the task is an inherent classification problem. It predicts a definite outcome: whether the red or blue fighter wins.

Logistic regression provides a simple, interpretable approach to binary classification. Individual features contribute to the likelihood of a fighter winning, and it serves as a benchmark for evaluating improvement from other models.

Random forests capture nonlinear relationships and interactions between fighter attributes that logistic regression cannot. It is robust to noise, handles mixed feature types, and naturally estimates feature importance, making it valuable for understanding which characteristics influence fight outcomes.

XGBoost is a highly optimized ensemble method that handles complex structured tabular data like the UFC dataset. It was used to compare with Random Forests because both are tree-based ensemble methods, but represent two different strategies for improving model performance. XGBoost handles class imbalance effectively, models complex decision boundaries, and provides better predictive accuracy due to gradient boosting, regularization, and optimized tree construction.

5 Interpretation of Findings

https://github.com/felixg318/UFC-Data-mining