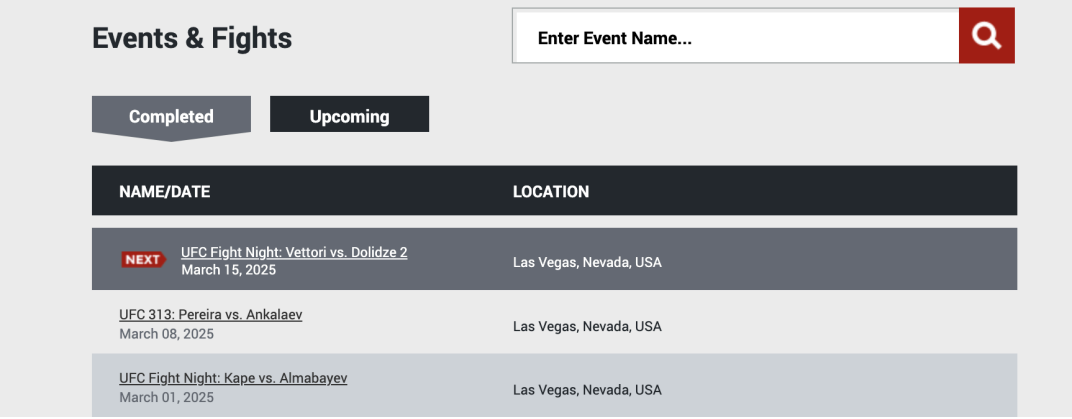
Data Analytics Final Project: Designing a UFC Predictive Model

Background

The Ultimate Fighting Championship (UFC) is the premier mixed martial arts (MMA) organization, featuring the top fighters competing across various weight classes. Founded in 1993, the UFC has evolved into a global phenomenon, showcasing elite fighters from diverse combat disciplines such as Brazilian Jiu-Jitsu, Muay Thai, wrestling, kickboxing, and more! Over the years, the sport has grown tremendously in popularity, drawing millions of viewers worldwide.

The UFC operates using weight classes, fight rules, and rankings to determine matchups and championship contenders. Fighters compete in five-minute rounds, with non-title fights lasting three rounds and title fights lasting five rounds. Bouts are officiated under official rules which regulate legal techniques, scoring criteria, and fouls. Judges score fights using a 10-point system, where the round winner typically receives 10 points, and the opponent receives 9 or fewer, based on striking, grappling, aggression, and control. Fighters progress through the UFC's ranking system, determined by a panel of media members. Championship fights and main events are the pinnacle of UFC competition, usually drawing millions of viewers.

This project is an extension of my final project for CS 512. For my CS 512 project, I created a UFC database by pulling data directly from UFC's website. Data was scraped using a Python script. The script gathered information on every completed UFC event card, title, date, and location.



I pulled the results overview for every card as well, this includes each fight, the outcome (finish), KD (knockdowns), STR (significant strikes), TD (takedowns), SUB (submissions), Weight Class, Method, Round, and Time.

W/L	FIGHTER	KD	STR	TD	SUB	WEIGHT CLASS	METHOD	ROUND	TIME
WIN	Manel Kape Asu Almabayev	0 0	61 40	0 0	0 0	Flyweight *PERF	KO/TKO Punches	3	2:16
WIN	Cody Brundage Julian Marquez	1 0	44 37	0 0	0 1	Middleweight	KO/TKO Punches	1	4:45
WIN	Nasrat Haqparast Esteban Ribovics	0 0	135 147	1 0	0 0	Lightweight *FIGHT	S-DEC	3	5:00
WIN	Hyder Amil William Gomis	0 0	115 101	1 0	0 0	Featherweight	S-DEC	3	5:00
WIN	Sam Patterson Danny Barlow	1 0	18 15	0 0	0 0	Welterweight	KO/TKO Punch	1	3:10
WIN	Mario Pinto Austen Lane	1 1	13 32	0 0	0 0	Heavyweight *PERF	KO/TKO Punch	2	0:39

Finally, I also attempted to pull details including the totals and significant strikes for every fight. Totals refer to KD, and SIG. STR and their respective percentages, Total STR(total strikes landed of strikes thrown), TD percentages, SUB. ATT (submissions attempted), REV(reversals, successfully escaping from a disadvantageous position), and CTRL (time a fighter spends in a dominant position on the ground or in a clinch). Significant Strikes contains SIG STR (landed of attempted) and its percentage and landed out of attempted for where strikes occur, (Head, Body, Leg, Distance, Clinch, and Ground).

TOTALS

FIGHTER	KD	SIG. STR.	SIG. STR. %	TOTAL STR.	TD	TD %	SUB. ATT	REV.	CTRL
Cody Brundage	1	44 of 79	55%	68 of 106	0 of 1	0%	0	0	0:48
Julian Marquez	0	37 of 73	50%	53 of 96	0 of 0	---	1	0	0:30

PER ROUND

SIGNIFICANT STRIKES

FIGHTER	SIG. STR	SIG. STR. %	HEAD	BODY	LEG	DISTANCE	CLINCH	GROUND
Cody Brundage	44 of 79	55%	39 of 71	1 of 2	4 of 6	34 of 65	5 of 8	5 of 6
Julian Marquez	37 of 73	50%	31 of 64	5 of 8	1 of 1	24 of 51	13 of 22	0 of 0

PER ROUND

The script then employed cleansing functions to format the scraped data properly and stored the data in a JSON format. The data was then uploaded to Google Big Query where it was stored and used to perform post hoc analysis.

Introduction

MMA is a highly unpredictable sport due to the numerous variables that contribute to fight outcomes. By analyzing historical fight data, we can attempt to identify key performance indicators that influence these outcomes. Sports betting has become increasingly popular, and the UFC is a favored sport for betting enthusiasts. While I am not a major participant in sports betting, I saw this project as a fun challenge to develop a model that could aid in UFC betting predictions.

My initial project focused on identifying a way to scrape and store UFC statistics as well as answer post-hoc questions. The goal now is to create a predictive model to enhance sports betting analysis for the UFC. Specifically, I aim to achieve a 70% accuracy rate in predicting fight winners. To accomplish this, I will implement and compare three different models:

1. **Logistic Regression**
2. **Random Forest**
3. **XGBoost**

The primary objective of this analysis is to predict fight winners based on statistical features derived from past UFC bouts. Using various metrics such as significant strike differentials, takedown effectiveness, reach differences, and betting odds, we aim to uncover patterns that contribute to victory. This study will not only focus on predictive modeling but also explore the interpretability of key variables influencing fight outcomes. Given the complex and non-linear nature of MMA fights, I do not expect Logistic Regression to perform well, as it will struggle with the complex interactions that play into the outcomes. The fight environment involves dynamic exchanges, varying styles, and situational factors, which are difficult to capture with a simple linear model.

Initially, I expected XGBoost to be the best-performing model, given its ability to handle feature interactions, capture non-linear patterns, and optimize decision boundaries through boosting techniques. XGBoost has been widely recognized for its success in structured data problems, making it a strong candidate for accurately predicting UFC fight outcomes. I also considered that Random Forest could perform well, as its approach is effective at reducing variance and handling a mix of categorical and numerical variables. This analysis will compare these models to determine which method provides the best combination of predictive power and interpretability for UFC fight predictions.

Data

Overview:

For this analysis, I utilized a dataset containing historical UFC fight records, including fighter statistics, fight metrics, betting odds, and bout details. Initially, I built a scraping tool to

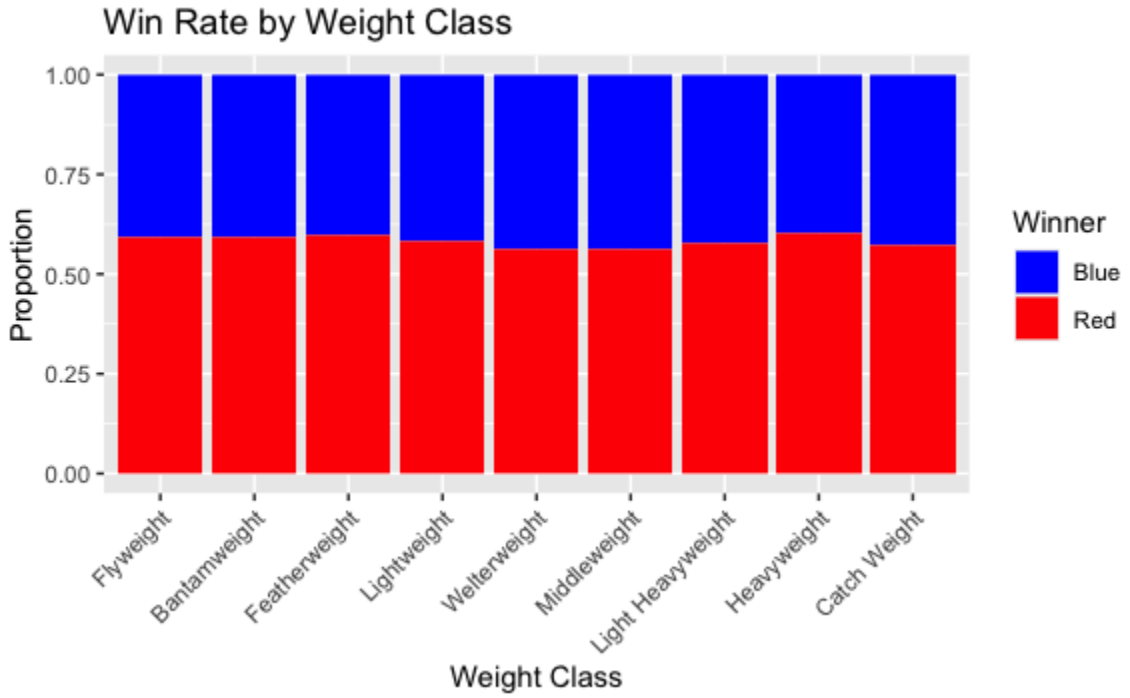
gather UFC data, but after some adjustments, I found an updated and cleaner dataset from an external source: Ultimate UFC Dataset. This dataset provided more structured and comprehensive information, making it a better foundation for predictive modeling. The dataset originally contained 117 features across 5,727 fights. Given the complexity of MMA, I expected results to be influenced by weight class and gender. To mitigate variability and improve model reliability, I decided to focus solely on men's UFC fights.

Key components of the dataset include:

- **Fighter Attributes:** Reach, height, weight, stance, and fight history (wins, losses, streaks).
- **Fight Metrics**: Significant strikes landed, takedown attempts, submission attempts, and control time.
- **Betting Odds**: Opening and closing odds for each fighter, including implied probabilities.
- **Fight Outcome**: Winner, method of victory, and round details.

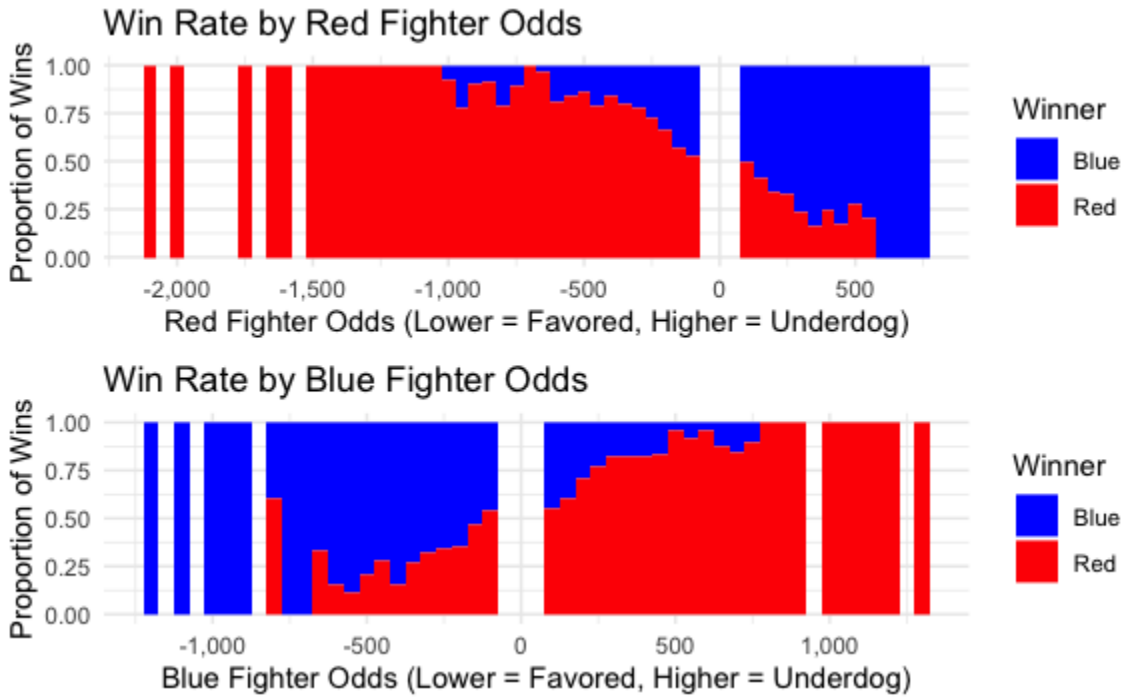
Exploratory Data Analysis:

To facilitate analysis, categorical variables were converted into factors. These include: Winner (Red or Blue Corner), Weight Class (Various UFC weight divisions), Fighter Stance (Southpaw, Orthodox, Switch, etc.), Better Rank (Comparative ranking between fighters), Finish Type (KO/TKO, Submission, Decision, etc.)

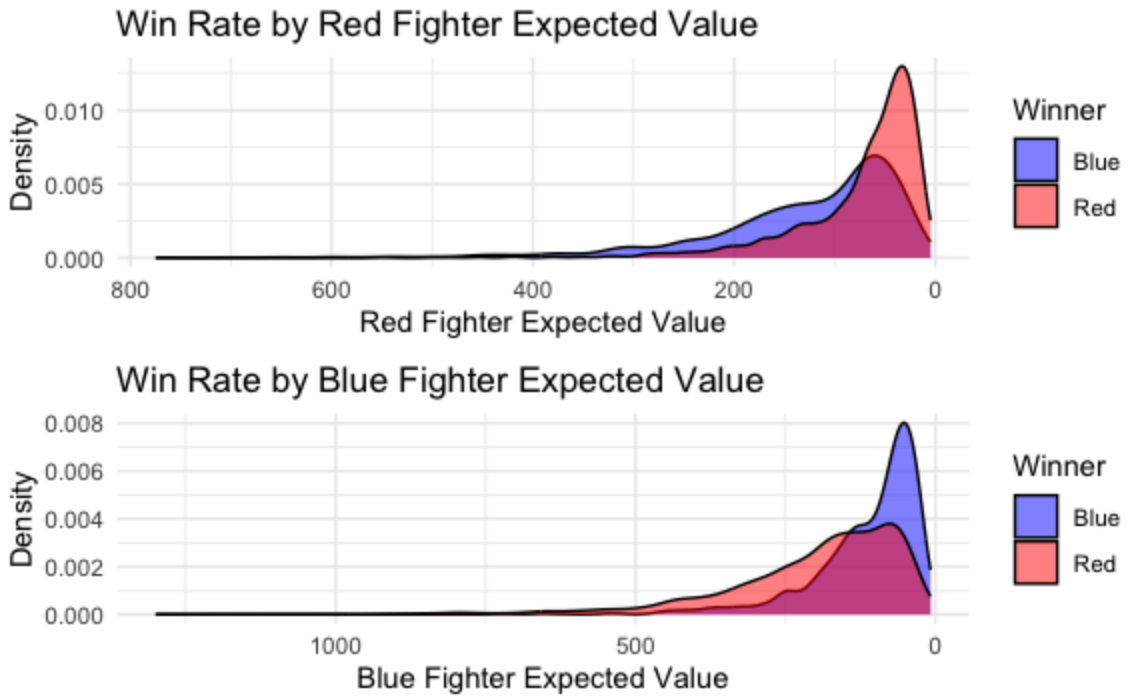


- Win rate distribution is relatively even among weight classes.

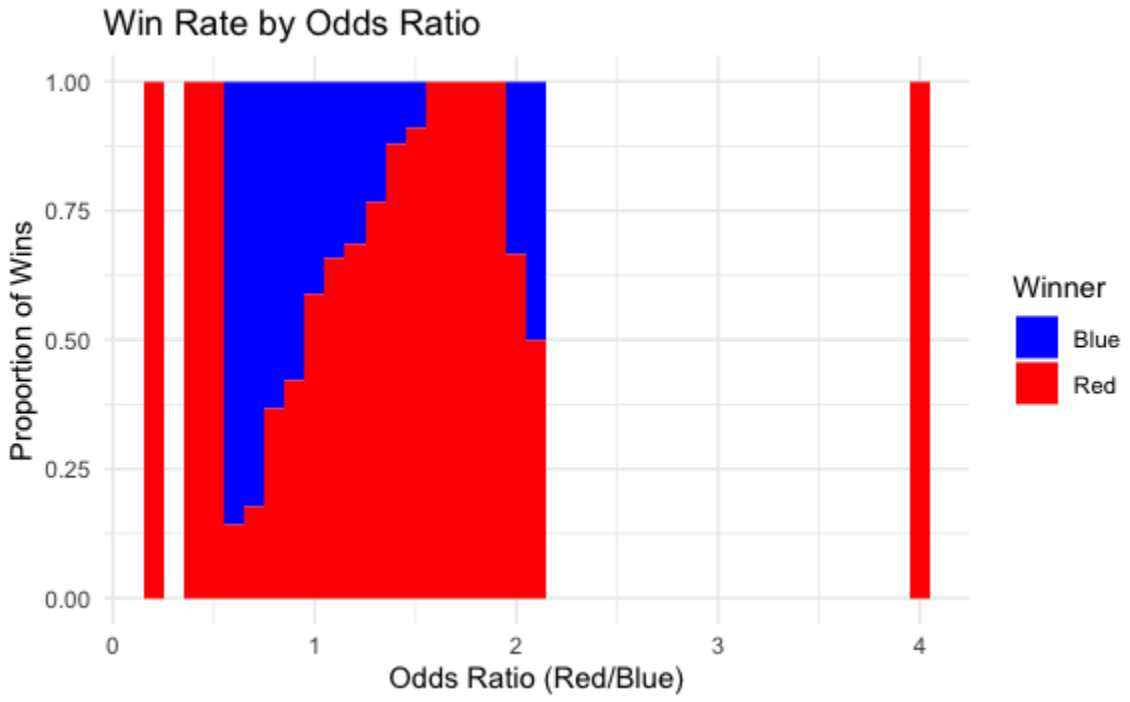
- Red corner fighters tend to win more often, aligning with the tradition of placing favored fighters in the red corner.



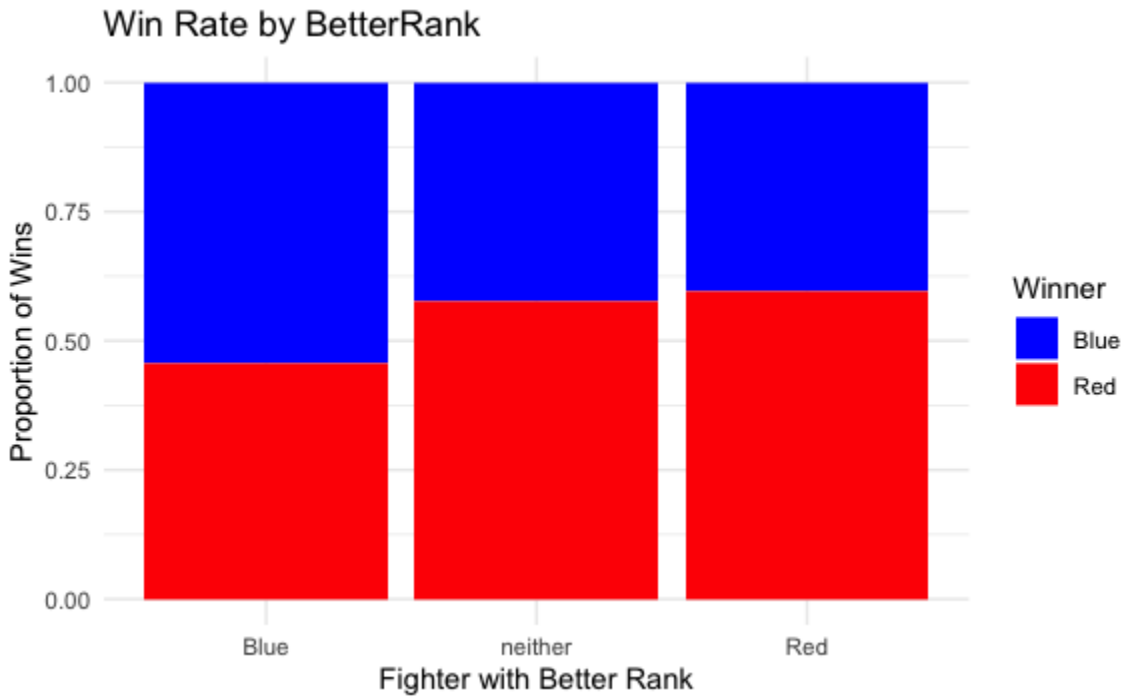
- Odds are a strong predictor of fight outcomes.
- Negative odds indicate a favored fighter, while positive odds indicate an underdog.
- Win probability follows a diagonal trend, with fluctuations in fights where odds are closely matched.



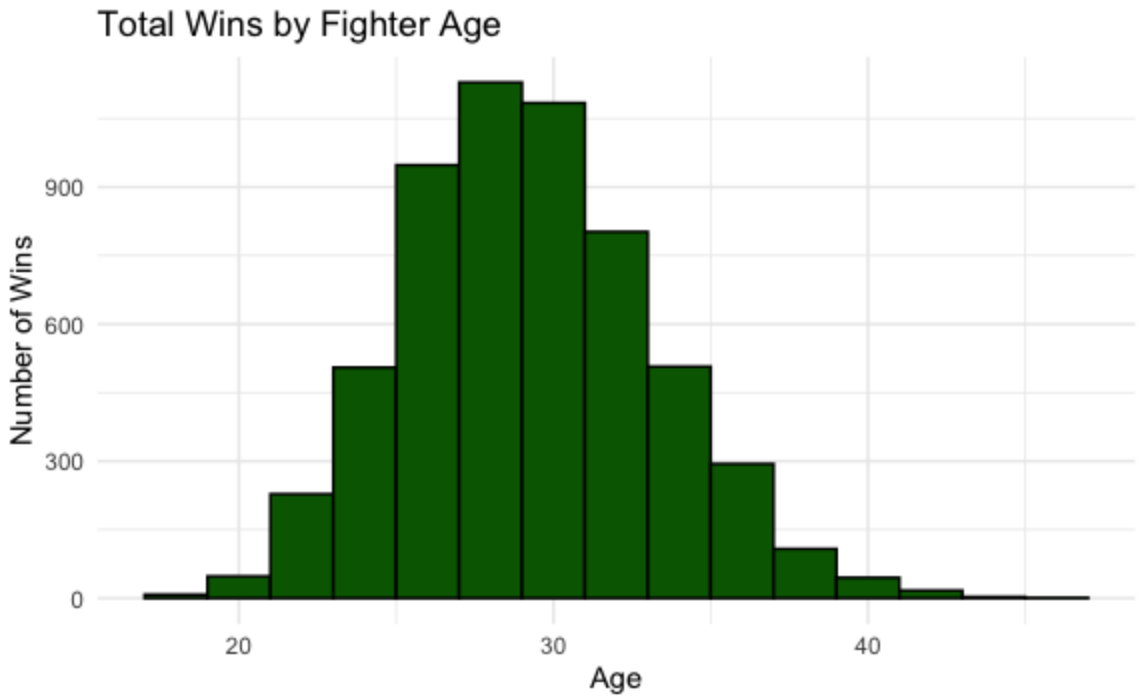
- Lower EV indicates a favored fighter, while higher EV suggests an underdog.
- Fights with higher EV tend to be less predictable.
- Blue corner EV extends further, reinforcing the trend of red corner fighters winning more frequently.



- If Odds Ratio > 1, the Red corner fighter was the underdog.
- If Odds Ratio < 1, the Blue corner fighter was the underdog.
- The odds ratio is a strong predictor, consistently outside the 1.2 to 1.5 bounds.



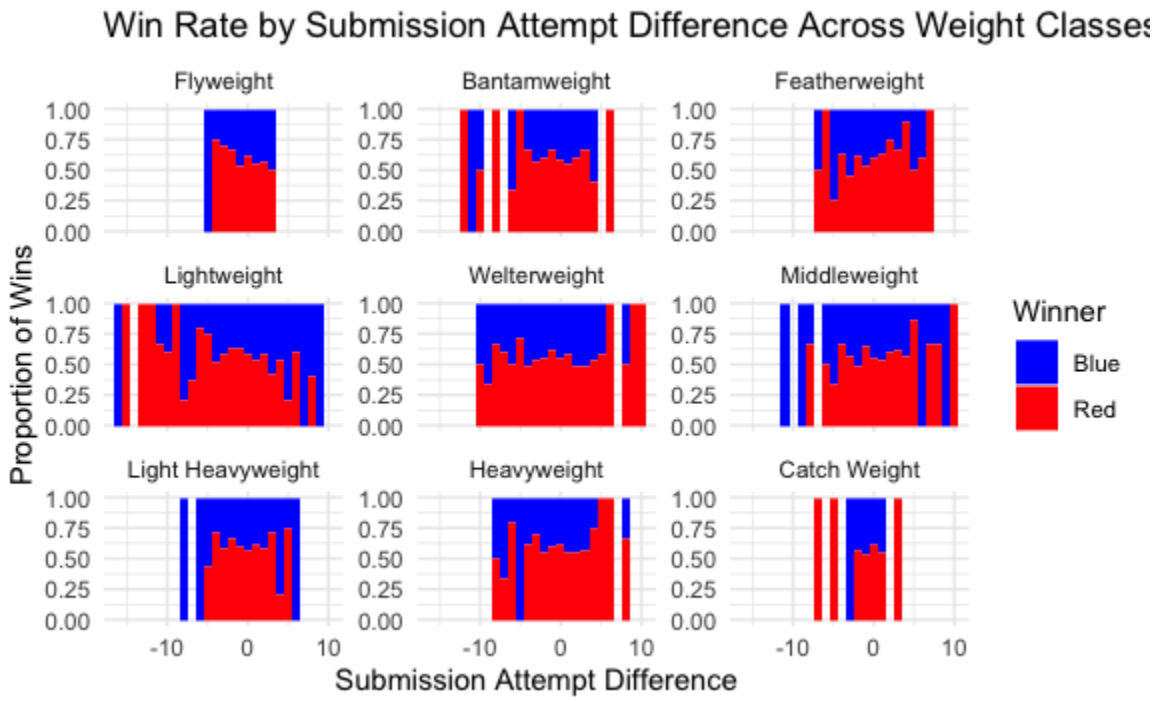
- When Blue fighters have a better rank, they win approximately 55% of the time.
- When ranking is undetermined, Red fighters win around 55% of the time.
- When Red fighters have a better rank, they win around 60% of the time.



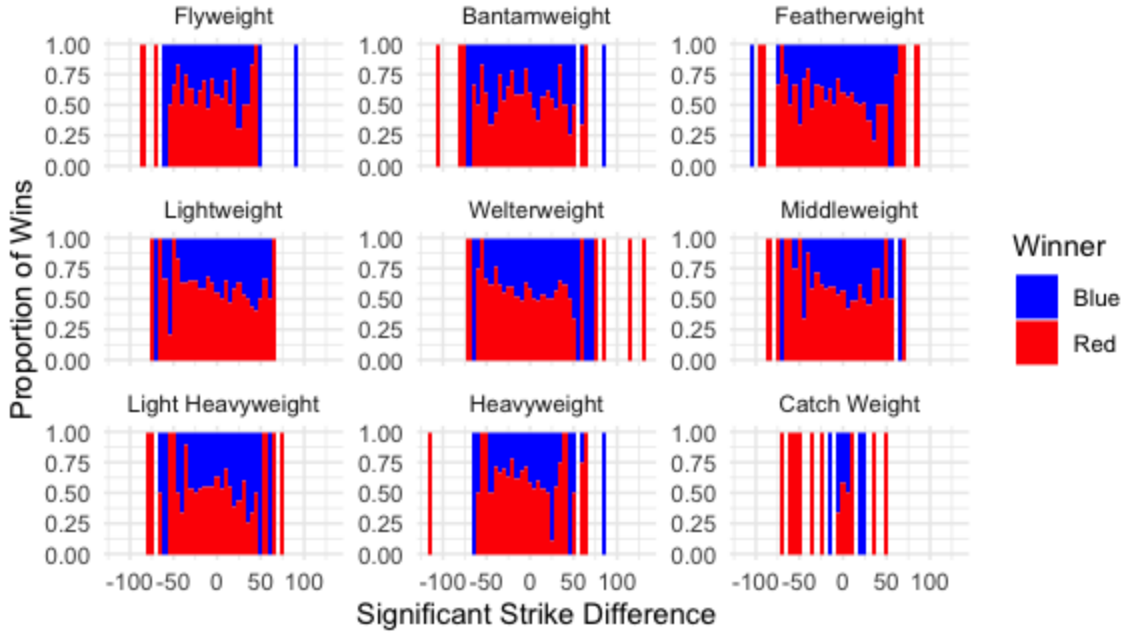
- Wins by age are approximately normally distributed, with the mean winning age around 30.
- This aligns with common knowledge that fighters typically peak in their early 30s due to experience and physical conditioning.

Win Rate by Fight Statistic Differences

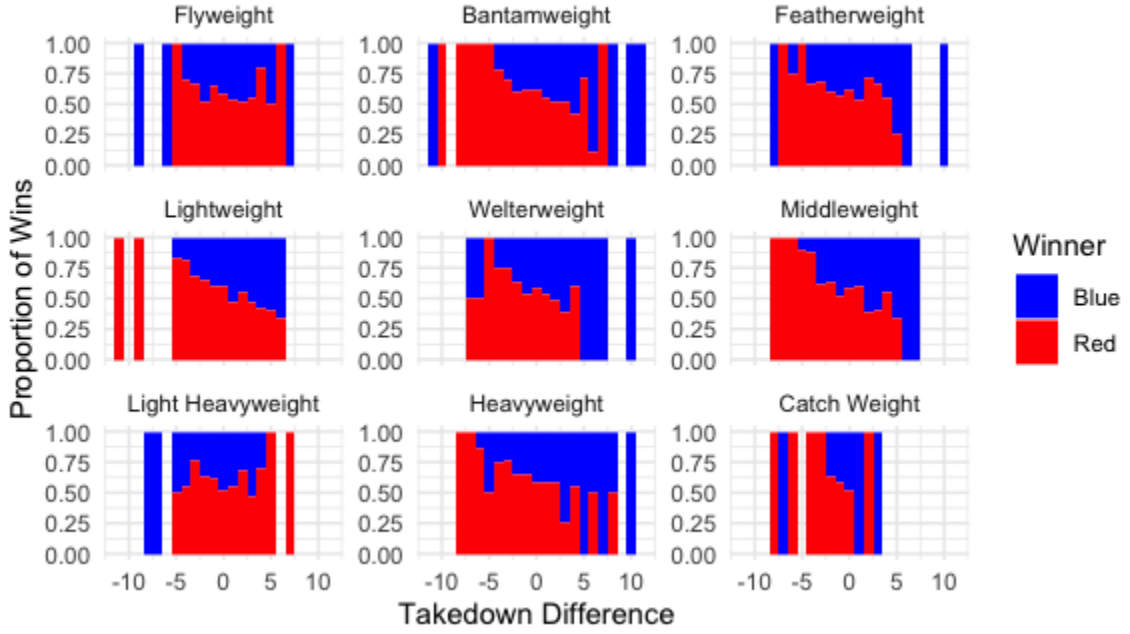
Analysis of differences in KO/TKO, significant strikes, takedowns, and submissions provides additional insights into fight outcomes.

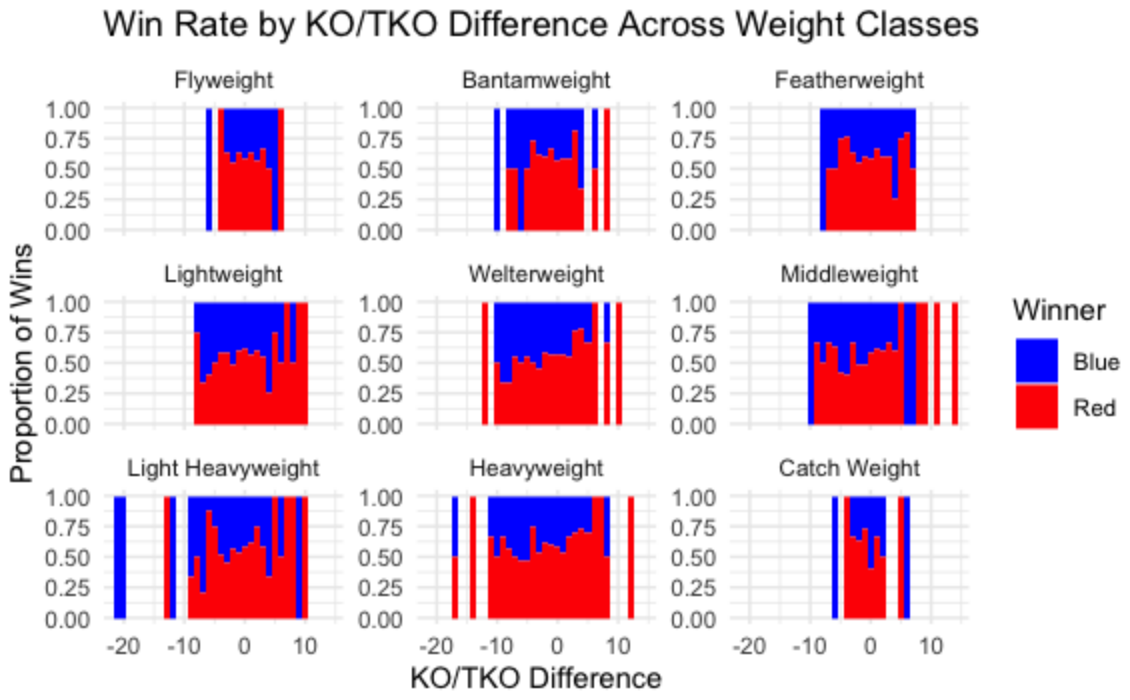


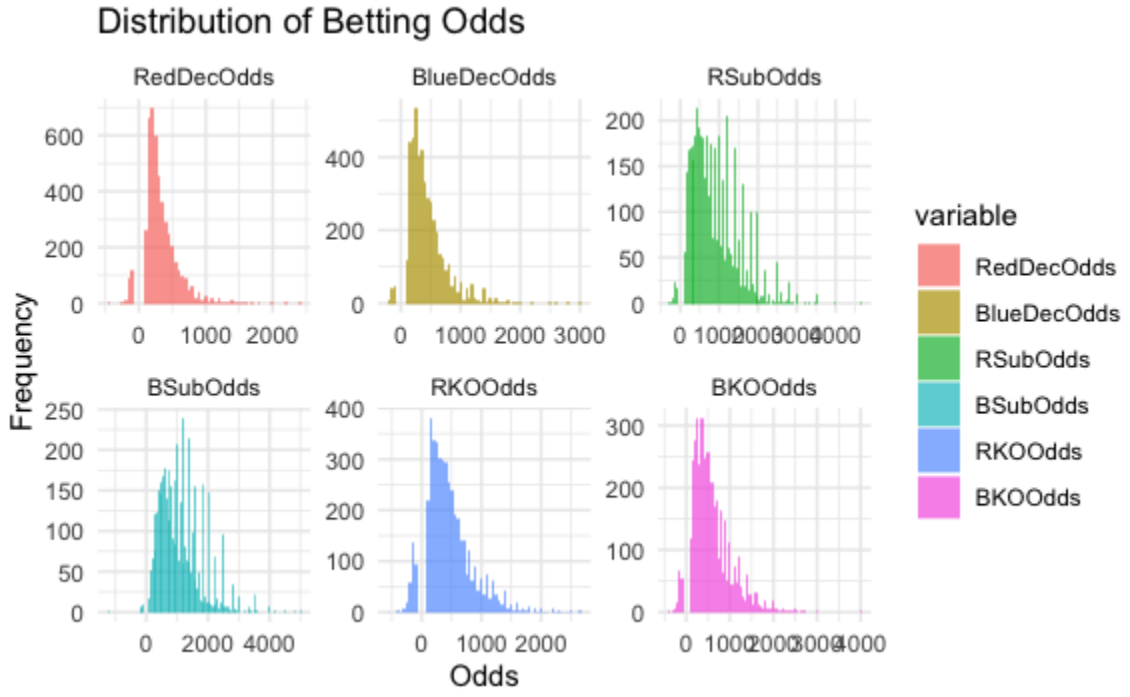
Win Rate by Significant Strike Difference Across Weight Classes



Win Rate by Average Takedown Difference Across Weight Classes







- A detailed visualization of odds distributions further supports the impact of betting markets on fight outcomes..

Data Cleaning and Preprocessing

Before analysis, several preprocessing steps were undertaken to ensure data quality.

Handling Missing Values:

Ranking variables had a high number of missing values, so all ranking-related columns were removed. Missing values in continuous variables were imputed using the median to handle skewness.

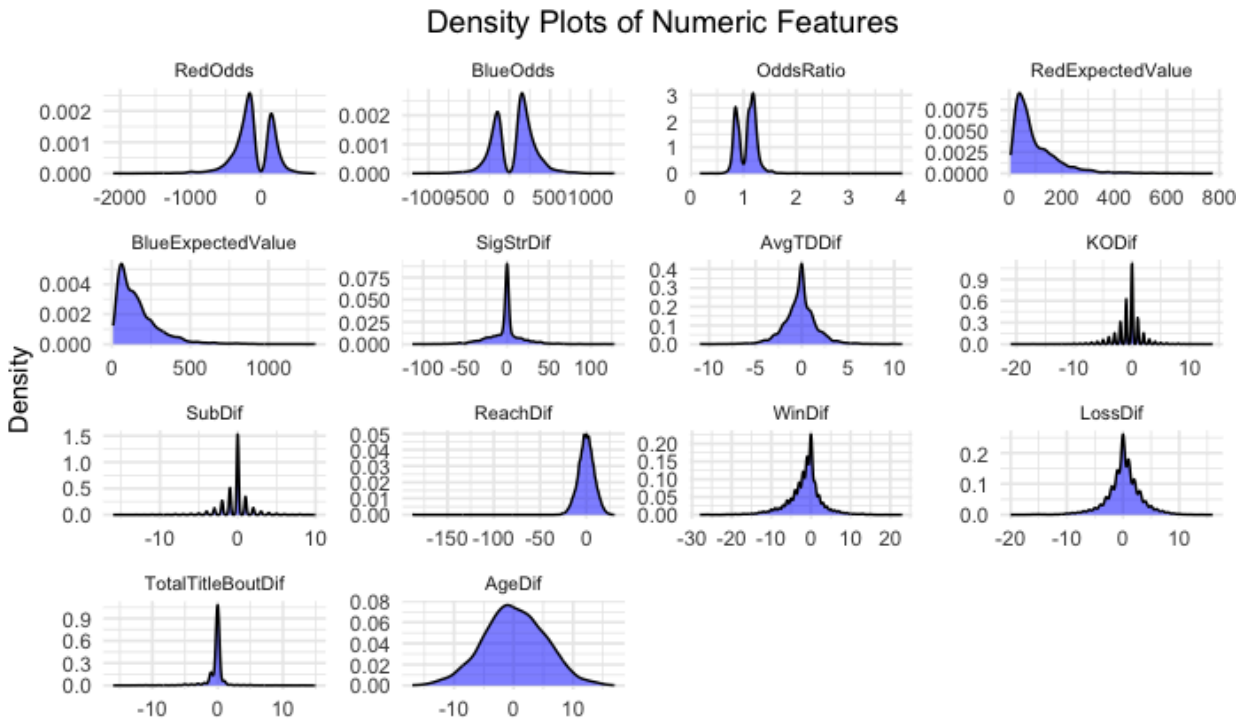
Categorical variables were imputed using mode imputation to preserve consistency.

A missing value table with percentages was generated to track data completeness.

Feature	Missing Count	Percent NA	Recommended Imputation
BlueAvgSigStrLanded	818	14.29	Median (Skewed Data)
BlueAvgSigStrPct	679	11.86	Mean or Median
BlueAvgSubAtt	736	12.86	Median
BlueAvgTDLanded	737	12.87	Medianfixed_nas
BlueAvgTDPct	744	13.0	Mean (if normally distributed)
RedAvgSigStrLanded	394	6.88	Median
RedAvgSigStrPct	312	5.45	Mean or Median
RedAvgSubAtt	312	5.45	Median
RedAvgTDLanded	312	5.45	Median
RedAvgTDPct	320	5.59	Mean or Median
TotalFightTimeSecs	517	9.03	Median
RedDecOdds	996	17.4	Median
BlueDecOdds	1025	17.9	Median
RSubOdds	1231	21.5	Median
BSubOdds	1255	21.92	Median
RKOOdds	1229	21.47	Median
BKKOOdds	1256	21.94	Median
RedOdds	169	2.95	Median
BlueOdds	167	2.92	Median
RedExpectedValue	169	2.95	Median
BlueExpectedValue	167	2.92	Median

OddsRatio	177	3.09	Recalculate
EVRatio	177	3.09	Recalculate
LogOddsRaatio	177	3.09	Recalculate
LogEVRatio	177	3.09	Recalculate
Log_OddsRatio	177	3.09	Recalculate
Log_RedExpectedValue	169	2.95	Recalculate
Log_BlueExpectedValue	167	2.92	Recalculate
FinishDetails	3087	53.92	Remove
FinishRound	517	9.03	Mode
FinishRoundTime	517	9.03	Mode

Feature Transformations:
Plotted distributions of numerical variables and applied log transformations where necessary to correct skewness.



- Log transformations were applied to:
  Odds Ratio, Red Expected Value (EV), Blue Expected Value (EV), Reach Difference, Total Title Bout Difference.

Feature Engineering:

- Divided decision wins into one unified group for simplification.
- Removed the FinishDetails column due to 56% missing values, making it unreliable for analysis.
- Removed redundant transformed features that did not contribute additional information to predictive models.

Addressing Multicollinearity:

Correlation analysis was conducted to eliminate redundant variables and ensure better model stability. By ensuring the dataset is clean, well-structured, and informative, I can maximize the accuracy and interpretability of the predictive models. The adjusted data set was then saved to use for analysis.

Models and Methodology

Logistic Regression

Assumptions for Logistic Regression

Assumption	Description	Potential Issue in Data	Mitigation Strategy
Linearity of Log-Odds	Predictors must have a linear relationship with log-odds.	Fight outcomes are influenced by non-linear interactions. Odds and Expected value could pose issues.	Applied log transformations to features like Odds Ratio and Expected Value.
Independence of Observations	Fights are considered independent events.	Fighters may improve or decline over time, based on (experience, age, mental, and injuries) meaning past fights can influence future results.	A more advanced model can incorporate a momentum feature given time series data.
No Perfect Multicollinearity	Predictors should not be perfectly correlated.	Some features are highly correlated.	Used VIF analysis to remove multicollinear features.
Large Sample Size	A large dataset is needed for stable coefficient estimates.	Weight classes have different balances, some disproportional.	Focused on men's UFC results to reduce variability.
NO Extreme Outliers	Outliers should not disproportionately affect predictions.	Extreme cases can skew results.	Applied feature scaling and removed extreme outliers.
Balanced Classes	Model assumes even distribution of outcomes.	There are more red corner wins than blue corner wins.	Used downsampling and class weighting to address the imbalance.

A logistic regression model was developed to predict fight winners. To address multicollinearity, a linear model was fitted, and aliased coefficients were removed. The alias function was used to identify perfectly correlated variables by fitting a linear model with all remaining variables. If a predictor is a perfect linear combination of other predictors, it is considered an "aliased coefficient" and does not provide independent information. These

aliased variables were removed to ensure numerical stability and prevent redundancy in the model.

Variance Inflation Factor (VIF) analysis was conducted to detect multicollinearity, which occurs when predictor variables are highly correlated. VIF measures how much the variance of a regression coefficient increases due to collinearity. A high VIF (typically above 5) suggests that a predictor is strongly correlated with others in the model, making it difficult to isolate its effect. Features with VIF values greater than 5, such as BlueLongestWinStreak and RedReachCms, were removed to improve model stability.

The dataset was split 80/20 into training and testing sets, and all numerical features were standardized using the means and standard deviations from the training set to improve model convergence. Standardization ensures that variables with different scales do not disproportionately influence the model.

The initial logistic regression model, trained with all predictors, achieved a residual deviance of 5603.3 and an AIC of 5641.3. However, accuracy was not computed at this stage due to the large number of variables. To improve performance, the model was reduced to include only the most statistically significant predictors. The refined model achieved an accuracy of 59.3%, with a precision of 0.59, a recall of 0.99, and an F1 score of 0.74. These results indicated that while the model was capturing most red corner wins, it was struggling with blue corner predictions, leading to class imbalance issues.

To address this, class balancing techniques such as upsampling and downsampling were tested. Upsampling, which involves increasing the number of observations in the minority class (blue corner wins) by randomly replicating existing examples, was attempted first. However, this resulted in a higher residual deviance of 6584.7 and a worse AIC of 6730.7, suggesting overfitting without improving predictive performance.

The strongest predictors in the logistic regression model include:

Logistic Regression - Most Significant Predictors

Feature	Significance	Explanation
Log_EVRatio	Strong ($p < 0.001$)	Betting odds strongly influence fight outcomes, aligning with market expectations.
BlueDecOdds	Strong ($p < 0.01$)	Higher blue corner odds indicate a lower chance of winning.
RedDecOdds	Strong ($p < 0.01$)	Higher red corner odds indicate a lower chance of winning.
RedAvgSigStrPct	Strong ($p < 0.05$)	Higher striking accuracy for red corner fighters increases the probability of winning.
BlueAvgSigStrPct	Strong ($p < 0.05$)	Higher striking accuracy for blue corner fighters also increases win probability.
RedWinsByDecisionSplit	Moderate ($p < 0.05$)	Decision wins suggest previous fight experience and tactical advantages in close fights.
BlueCurrentWinStreak	Moderate ($p < 0.05$)	Winning streaks indicate strong momentum and a higher chance of continued success.
RedAge	Moderate ($p < 0.05$)	Younger red corner fighters have a slightly higher probability of winning.
BlueAge	Moderate ($p < 0.05$)	Younger blue corner fighters also show increased chances of winning.

Logistic Regression determines feature significance using:

- **Coefficient Size**: The larger the coefficient, the stronger the feature's influence on win probability.
- **P-values**: Predictors with **$p < 0.05$** are considered **statistically significant** (stronger evidence that they impact the outcome).
- **Odds Ratios**: The exponentiated coefficient (`exp(coef)`) shows how much the probability of winning changes for a one-unit increase in that predictor.
- A positive coefficient means higher values of the predictor increase the probability of a red-corner win.
- A negative coefficient means higher values of the predictor increase the probability of a blue corner win.
- An odds ratio > 1 means the feature increases the likelihood of a red corner win.

- An odds ratio < 1 means the feature decreases the likelihood of a red corner win (favoring a blue corner).
- Betting odds (Log_EVRatio, BlueDecOdds, RedDecOdds) were the strongest predictors, reinforcing that fight outcomes closely align with market expectations.
- Striking accuracy (RedAvgSigStrPct, BlueAvgSigStrPct) had a strong effect on predicting winners, meaning that fighters with higher striking accuracy had a significantly higher chance of winning.
- Experience factors (BlueCurrentWinStreak, RedWinsByDecisionSplit) suggested that fighters with a history of consistent wins or decision victories were more likely to continue winning.
- Age factors (RedAge, BlueAge) indicated that younger fighters had a higher probability of winning, aligning with trends in endurance and reaction speed.

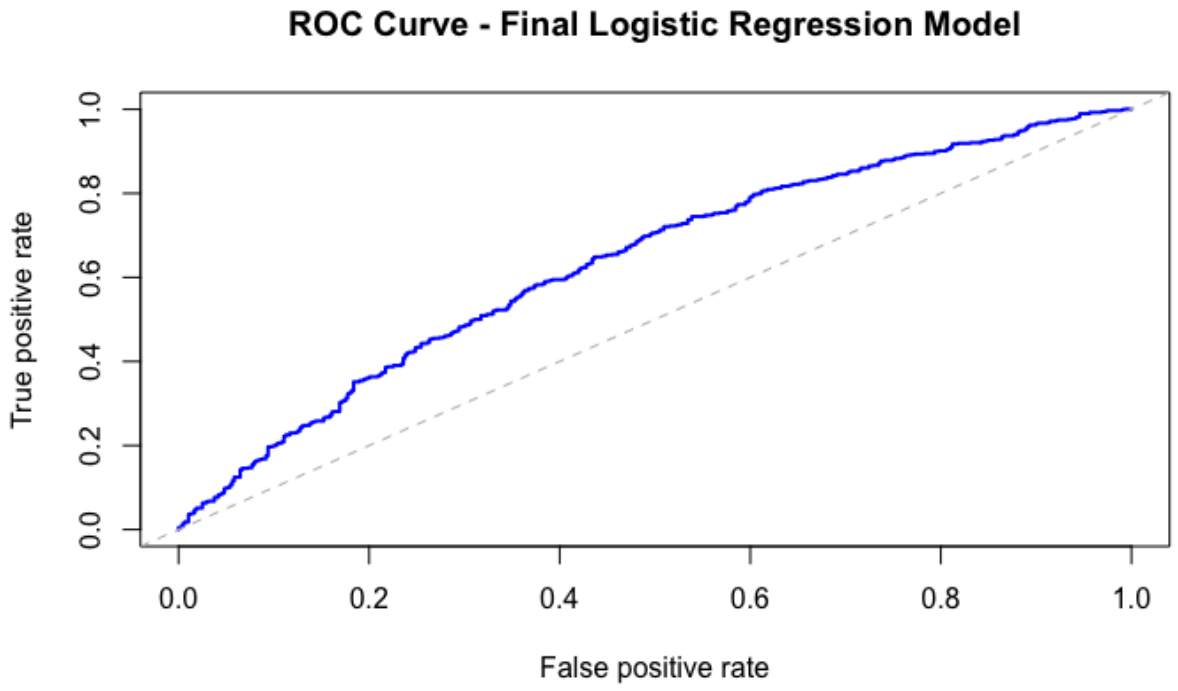
The final model was trained on a balanced dataset using downsampling, where the number of red corner fights was reduced to match the number of blue corner fights. This resulted in a residual deviance of 4712.8, an AIC of 4858.8, and an improved accuracy of 62.79%. The model still exhibited a slight bias toward predicting red-corner wins, but the class balancing helped mitigate this issue.

In the final evaluation, the best-performing model achieved an accuracy of 62.97%, with a precision of 0.65, a recall of 0.80, and an F1 score of 0.72. Precision measures the proportion of correct positive predictions out of all predicted positives, meaning that when the model predicted a red corner win, it was correct 65% of the time. Recall measures how many actual positive cases were correctly identified, indicating that the model correctly identified 80% of actual red corner wins. The F1 score, which balances precision and recall, reflected a reasonable trade-off between correctly predicting red-corner wins and avoiding false positives. The AUC for the final model was .6347, meaning its ability to distinguish between red and blue wins leans toward the weaker side but is better than a random guess. Cross-validation was attempted, but my computer lacked the processing power required.

Despite improvements, the model continued to show a slight bias toward red-corner fighters, though class balancing helped reduce the imbalance. Betting odds remained strong predictors of fight outcomes, reinforcing the influence of market expectations in UFC fights.

Logistic Regression Model Comparison

Metric	Initial Model (reduced)	Final Model (down sampling)
Residual Deviance	5603.3	4712.8
AIC	5641.3	4858.8
Accuracy (%)	59.3	62.97
Precision	.59	.65
Recall	.99	0.8
F1 Score	.74	.72



The ROC curve shows that my final model has better predictive power than chance but struggles with fights that have close odds.

Random Forest

Assumptions for Random Forest

Assumption	Description	Potential Issue in UFC Data	Mitigation Strategy
No Linearity Requirement	Handles non-linear relationships.	It works well for complex fight interactions compared to logistic regression.	NA
Handles Multicollinearity	Can handle multicollinearity	Some features are highly correlated.	Used feature importance analysis to remove redundant features.
Independence	Features don't need to be independent.	NA	NA
Sufficient Sample Size to Avoid Overfitting	Enough data provided to avoid overfitting.	Data set size is large, and some weight classes have fewer fights.	removed catchweight.
Class Balance Helps Improve Performance	Handles imbalance better than logistic regression but still struggles with imbalance.	Red corner fighters win more often, creating bias.	Used class weighting and engineered additional features to improve balance.

A Random Forest model was trained using numerical features. It is expected that the Random Forest model will outperform the logistic regression model as it is better suited for this task because it should better capture complex interactions and be less prone to overfitting. Data was prepared by using a correlation matrix to identify highly correlated pairs of features to remove, to reduce redundancy. Irrelevant features like Name, Location, and Date were removed and categorical features were converted to factors.

A baseline model was established, using all remaining features, trained on a tree size of 500. This model resulted in Out-of-Bag (OOB) error estimate of 33.96%, with an initial test accuracy of 65.67% already outperforming the logistic regression model and setting the benchmark for improvement. A class imbalance favoring red fighters is present. The most significant predictors include

Random Forest - Most Significant Predictors

Feature	Importance Score	Explanation
Log_EVRatio	234.1	Betting odds remain the most influential feature.
Log_OddsRatio	173.1	Odds ratios reveal the relative likelihood of one fighter winning over another.
BlueDecOdds	115.9	Higher blue corner odds indicate a lower probability of winning.
RedDecOdds	100.7	Higher red corner odds indicate a lower probability of winning.
RedAvgSigStrPct	97.4	Striking accuracy remains highly significant for predicting fight outcomes.
BlueAvgSigStrPct	97.1	Blue fighter striking accuracy is nearly as influential as for red fighters.
RedAvgTDPct	88.8	Takedown percentage significantly impacts fight control and scoring.
BlueAvgTDPct	74.9	Blue fighter takedown ability is crucial for controlling fights.
ReachDif	68.4	Reach differences impact striking effectiveness, especially in stand-up fights.
AgeDif	78.3	Age difference remains a relevant factor, affecting endurance and durability.

Random Forest determines predictor importance using:

- **Mean Decrease in Gini Impurity:** Measures how much a feature improves decision splits within the trees.
- **Feature Contribution Across Trees:** Averages the importance of each feature over multiple decision trees.

Higher feature importance values indicate stronger predictive power. Random Forest captures complex interactions better than Logistic Regression because it does not assume a linear relationship.

- The most important features (Log_EVRatio, Log_OddsRatio, RedAvgSigStrPct, etc.) contributed significantly to reducing prediction errors.

Betting odds were still dominant predictors, but striking accuracy, takedown accuracy, and reach difference had significantly more influence than in Logistic Regression.

- Weight class was not as important in Random Forest, suggesting that individual fighter skills matter more than division-based trends.

Feature importance analysis was performed using MeanDecreaseGini and the top 15 important features were preserved. A grid search was used to tune the model's hyperparameters. The parameters of interest include the number of variables per split (mtry), minimum node size (nodesize), and number of trees (ntree). The best model resulted in a tree size of 500, 4 variables per split, and node size of 3. Accuracy for the tuned model resulted in 66.26 %, a .59 % increase over the baseline model.

In order to address the bias of red corner wins, three new features were constructed.

- BlueAggression = (BlueAvgSigStrPct + BlueAvgTDPct) / 2
- BlueExperience = BlueWinsByDecisionUnanimous + BlueWinsByKO + BlueWinsBySubmission
- BlueUnderdogScore = BlueDecOdds / RedDecOdds (higher values indicate a larger underdog)

A weighted class approach was used (classwt = c(2,1)) in order to provide blue corner fighters with greater influence. This final approach resulted in a test accuracy of 67.42%, a 1.75% increase in accuracy over the baseline model, and a 4.45% increase in accuracy when compared to the logistic regression final model. AUC value of .7227 puts its ability to discriminate at a moderate level, a solid improvement over the logistic regression model.

Random Forest Models

Metric	Baseline Model	Tuned Model	Final Model (new features)
OOB Error Rate	33.96	34.74	NA
Accuracy	65.67	66.26	67.42

XGBoost

Assumptions for XGBoost

Assumption	Description	Potential Issue in UFC Data	Mitigation Strategy
No Linearity Requirement	Can handle complex, non-linear relationships.	NA.	NA
Feature Independence	Works best if features are uncorrelated.	Some correlated features may reduce interpretability.	Used feature selection and correlation analysis.
Boosting Assumes Residual Patterns Can Be Learned	Each new tree should improve upon previous errors.	If the initial model is performs poorly, XGBoost might amplify errors.	Used early stopping and hyperparameter tuning to avoid overfitting.
Sensitive to Class Imbalance	More affected by class imbalance than Random Forest.	Red Blue Imbalance	Used scale_pos_weight adjustment to balance class distributions.
Requires More Data for Generalization	Works best with large datasets to prevent overfitting.	UFC fights may be relatively small-scale, especially features that lack information.	May need to expand data by using more granular dataset.

An XGBoost model was trained using numerical features. XGBoost is a gradient-boosting algorithm that sequentially builds trees, learning from previous errors to refine predictions. It was expected that XGBoost would perform competitively with Random Forest by capturing complex feature interactions while maintaining regularization to prevent overfitting.

Data preparation involved removing highly correlated features identified through a correlation matrix to reduce redundancy. Non-informative columns such as fighter names, locations, and dates were also removed. Categorical features were converted into a numerical format, and the target variable, Winner, was encoded as a binary classification (0 = Blue, 1 = Red) to align with XGBoost's requirements. The dataset was split into training and testing sets using an 80/20 split.

A baseline XGBoost model was trained using an initial parameter set, including a maximum tree depth of 6, a learning rate (eta) of 0.3, and a subsampling rate of 0.8. The model was trained for 100 boosting rounds with early stopping after 10 rounds if no improvement was observed. The baseline model achieved a log-loss of 62.23%, indicating moderate predictive capability. Feature importance was evaluated using XGBoost's built-in importance function, with the top 20 features retained for further tuning. Hyperparameter tuning was conducted using a grid search approach, refining key parameters such as the number of boosting rounds, learning rate, and tree depth. The best model configuration included a learning rate of 0.1, a tree depth

of 6, and 400 boosting rounds. Class imbalance was present again, in this model as well. The most significant predictors for the BGBoost model include

XGBoost - Most Significant Predictors

Feature	Importance Score	Explanation
Log_EVRatio	129.6	Betting odds remain the strongest feature, but slightly less influential than in Random Forest.
Log_OddsRatio	89.8	Odds ratios still contribute to predictions, though not as strongly as in tree-based models.
BlueDecOdds	63.4	Blue corner odds remain an indicator of underdog status and win probability.
RedDecOdds	57.7	Red corner odds remain important, indicating fight competitiveness.
WeightClass	62.1	Weight class plays a larger role in XGBoost, showing differences across divisions.
RedAvgSigStrPct	53.7	Striking accuracy remains a key indicator, though slightly lower in importance than in Random Forest.
BlueAvgSigStrPct	51.6	Blue fighter striking accuracy is still a relevant but weaker predictor.
RedAvgTDPct	56.1	Takedown percentage influences win probability but is less dominant than in Random Forest.
BlueAvgTDPct	51.6	Blue corner fighters with better takedown rates show improved win chances.
ReachDif	54.7	Reach advantage influences striking-based fights but less than expected.

XGBoost determines feature importance using:

- **Gain** – Measures how much a feature improves decision splits across boosting iterations.
- **Cover** – Represents how often a feature is used in splits across all boosting rounds.
- **Weight** – Shows the number of times a feature is selected in trees.

Features with the highest Gain have the strongest impact on predictions. XGBoost captures more refined relationships compared to Logistic Regression and Random Forest because it adjusts decision boundaries dynamically.

- XGBoost placed slightly more emphasis on weight class than Random Forest, suggesting that fight outcomes differ across divisions.

- Betting odds were still the dominant predictor, reinforcing that fight results align with bookmaker expectations.

To address the class imbalance favoring red-corner fighters, the model incorporated a scale_pos_weight adjustment to increase the influence of blue-corner fights. This resulted in a refined model with a test log-loss of 62.66, improving generalization compared to the baseline model.

Additional tuning was performed with an increased tree depth of 8, a reduced learning rate of 0.12, and refined subsampling parameters. The final optimized model was trained for 700 rounds with an early stopping set to 20 rounds. Probability thresholds were adjusted to optimize classification accuracy, with the best threshold identified at 0.415. This final model resulted in a test accuracy of 66.11%, outperforming the logistic regression model, and closely matching the best-performing Random Forest model. AUC = .7049, putting discrimination in the moderate range, improving upon the logistic regression model, and performing close to the random forest model.

XG Boost Comparisons

Metric	Baseline Model	Tuned Model	Final Model
Training Log-loss	.6747	.6223	.4564
Testing L	.681	.623	.6389
Final Test Accuracy	NA	NA	.6611
Blue Wins % Correct	NA	NA	64.24
Red Wins % Correct	NA	NA	68.96

Results

The performance of the three models, Logistic Regression, Random Forest, and XGBoost was evaluated based on accuracy, precision, recall, and AUC, with additional focus on the most significant predictors identified in each model. The Logistic Regression model, after feature selection and downsampling, achieved a final accuracy of 62.97% with an AUC of 0.6347. The model struggled with class imbalance, particularly in predicting blue corner wins. The final model had an F1 score of 0.72, balancing precision (0.65) and recall (0.80), but still exhibited bias toward predicting red corner wins. Betting odds emerged as the strongest predictor, with Log_EVRatio, BlueDecOdds, and RedDecOdds playing a dominant role in

determining outcomes. Striking accuracy (RedAvgSigStrPct, BlueAvgSigStrPct) was also a significant factor, suggesting that fighters with higher striking efficiency had a clear advantage. Experience-based features, such as RedWinsByDecisionSplit and BlueCurrentWinStreak, contributed to win probability, reinforcing that fighters with more success in previous bouts were more likely to win. Despite these insights, Logistic Regression's weak AUC score of 0.6347 indicates that it was limited in its ability to distinguish winners from losers beyond basic linear relationships. Cross-validation was attempted but could not be completed due to computational limitations.

The Random Forest model initially outperformed Logistic Regression with an accuracy of 65.76% when using all available features. Feature selection and hyperparameter tuning further improved the model, leading to a final accuracy of 67.42%. The introduction of three new features, BlueAggression, BlueExperience, and BlueUnderdogScore, along with class reweighting (favoring blue corner fights) helped address bias and improved predictive performance. Random Forest identified betting odds (Log_EVRatio: 234.1, Log_OddsRatio: 173.1, BlueDecOdds: 115.9, RedDecOdds: 100.7) as the most critical factors in predicting fight outcomes. However, it also emphasized striking accuracy (RedAvgSigStrPct: 97.4, BlueAvgSigStrPct: 97.1), takedown accuracy (RedAvgTDPct: 88.8, BlueAvgTDPct: 74.9), and reach difference (ReachDif: 68.4). These findings indicate that Random Forest was more capable than Logistic Regression in identifying key fight characteristics beyond betting markets. The AUC score of 0.7228, the highest among all models, suggests that Random Forest had the best ability to distinguish winners from losers using the given features.

The XGBoost model trained using a gradient boosting framework, refined predictions over multiple boosting rounds. The initial training log loss was 0.6223, and class balancing techniques further reduced the testing log loss to 0.6230. After adjusting for class imbalances and optimizing probability thresholds, the final model achieved an accuracy of 66.11%, slightly below Random Forest but outperforming Logistic Regression. XGBoost identified betting odds (Log_EVRatio: 129.6, Log_OddsRatio: 89.8, BlueDecOdds: 63.4, RedDecOdds: 57.7) as the strongest predictors, but its feature importance rankings showed slightly less emphasis on striking accuracy compared to Random Forest. Instead, weight class (62.1) emerged as a more influential predictor, suggesting that fighter performance differs significantly across divisions. While XGBoost handled class balancing more effectively than Logistic Regression, it surprisingly did not outperform Random Forest, achieving an AUC score of 0.7049, slightly lower than Random Forest's but still in the moderate discrimination range.

Comparison of Most Significant Predictors Across Models

Feature	Logistic Regression	Random Forest	XGBoost	Explanation
Log_EVRatio	Strong	**234.1**	**129.6**	Betting odds were the strongest predictor across all models.
Log_OddsRatio	-	**173.1**	**89.8**	Odds ratios were more important in Random Forest and XGBoost.
BlueDecOdds	Strong	**115.9**	**63.4**	Higher blue corner odds reduce win probability across models.
RedDecOdds	Strong	**100.7**	**57.7**	Higher red corner odds reduce win probability across models.
WeightClass	-	-	**62.1**	Weight class had a larger impact in XGBoost than in other models.
RedAvgSigStrPct	Strong	**97.4**	**53.7**	Striking accuracy was important in all models but strongest in Random Forest.
BlueAvgSigStrPct	Strong	**97.1**	**51.6**	Blue fighter striking accuracy had

Final Model Comparison

Metric	Logistic Regression	Random Forest	XGBoost
Final Test Accuracy %	62.97	67.42	66.11
AUC	.6347	.7228	.7049
Correct Predictions Blue %	58.73	62.91	64.24
Correct Predictions Blue %	64.58	77.19	68.96

Discussion

Despite demonstrating moderate predictive capability, none of the models achieved an accuracy above 70%, which suggests limitations in both data availability and model selection for UFC fight prediction. While the models performed reasonably well in predicting fight winners, several challenges and limitations were encountered throughout the project. One major challenge was the inherent unpredictability of MMA fights. The dataset captured historical statistics, but fight outcomes are influenced by numerous dynamic factors such as fighter injuries, game plans, and last-minute fight changes that are difficult to quantify in a structured dataset.

Another limitation was the class imbalance, with respect to blue corner wins. The bias toward red corner fighters winning was apparent due to the practice of placing favored fighters in the red corner. Various techniques such as upsampling and adjusting class weights helped mitigate this issue, but future work could benefit from more refined data balancing techniques.

The data itself posed limitations, specifically in its granularity. The dataset contained aggregated statistics per fight, but additional granularity, such as round-by-round data, could enhance predictive power. A model trained on detailed per-round performance data might better capture momentum shifts, fighter endurance, and strategic adjustments that play a crucial role in determining the fight winner. This would require a more sophisticated scraping method, I struggled with this in my initial project focused on data collection and storage.

Another significant challenge was computational limitations. Cross-validation, a common technique for improving model robustness, proved to be computationally expensive given the dataset size and feature complexity. My system struggled to handle repeated model training cycles efficiently, forcing me to opt for a simpler train-test split rather than employing cross-validation. Future implementations could benefit from leveraging cloud computing resources or more optimized model training pipelines to handle larger computational loads more effectively.

While it was expected for both the Random Forest model and XG Boost model to outperform the logistic regression model I was surprised to see that the Random Forest model

performed better than our XG Boost model. One point I found interesting was that weight class had less predictive importance in the Random Forest model but was more significant in XGBoost. This difference likely stems from how each algorithm processes feature interactions. Random Forest treats each decision tree independently and depends on majority voting. This is important because weight class does not consistently improve decision splits across all trees. Therefore, fighter-specific attributes such as reach, striking accuracy, etc. were more influential in predicting winners. XGBoost builds trees sequentially and learns from residual errors, making it more sensitive to interactions between features; important for complex predictions like this. We could assume that weight class does not directly determine fight outcomes but instead modifies the way other features impact the result. This is intuitive, an example would be striking accuracy may be more relevant in lighter divisions where speed dominates and knock-out power is less, while KO power could be a stronger predictor in heavier divisions. Exploring this further could help to clarify how weight class interacts with other fight statistics in predictive modeling. While this information appears crucial, the sample size could be responsible for why the Random Forest Model performed better, XGBoost may have overemphasized certain fight-specific interactions, leading to potential overfitting. In addition, Since betting odds were already a strong predictor, XGBoost may have become too reliant on these features, while Random Forest distributed feature importance more evenly resulting in a more stable and reliable model.

One of the strongest predictors in our models was betting odds, specifically the features Log_EVRatio, BlueDecOdds, and RedDecOdds. This aligns with expectations, as betting odds are derived from extensive market analysis, historical fight data, and expert opinions. They reflect both objective fight statistics and subjective factors such as public perception, betting trends, and insider information. Given that sportsbooks adjust odds based on large-scale betting behavior, their predictive power is naturally high. However, this reliance on odds also introduces potential bias, popular or hyped fighters may have inflated odds due to public image rather than actual statistical information. To determine the true value of betting odds in our model, future work could compare results by training a model without odds-based features to assess whether statistical fight data alone provides comparable predictive power.

During my research, I found a claim that stated:

"If the odds don't make sense to you, the market knows something you don't. The public (and the market) is still susceptible to hype, but overwhelmingly, massive favorites win over 90% of the time no matter what else you measure against them."

This suggests that a market-inclusive analysis will almost always predict favorites to win, reinforcing the accuracy of betting markets but failing to provide an independent statistical analysis of fight performance. To improve prediction accuracy and identify potential inefficiencies in betting trends, a market-excluded model should be developed. Removing odds-related features (Log_EVRatio, BlueDecOdds, RedDecOdds) would allow us to:

- Determine whether fight statistics alone (e.g., striking, takedown accuracy, reach) can accurately predict winners.

- Identify overhyped fighters, where the market's prediction differs from statistical reality.
- Find better betting opportunities, where a fighter may be an overwhelming favorite but historical performance metrics suggest a much closer fight.

A key path for improvement is model selection. While logistic regression, random forest, and XGBoost provided valuable insights, incorporating a more complex deep learning approach such as a Multilayer Perceptron (MLP) or Transformer-based model could potentially yield better results. These models require more extensive data collection, but they could capture nuanced fight dynamics and temporal trends better than the models I selected. Ultimately, while the models demonstrated meaningful predictive capabilities, there is room for improvement in feature engineering, dataset expansion, and model complexity to better capture the unpredictable nature of UFC fights.