## **UFC: Red vs. Blue**

## *Predicting Fight Outcomes using Logistic Regression with resampling techniques*

## Group 5

## 

## 

## 

## By Jackson Windhorst, Jeet Shah, Jordan Allwood, Shraavani Bekkary, Matthew Nighting-Henderson, Meghana Mukkoti, Shivani Ananth, Ganesh Ravindran, Baylor Dalsemer, Raelyn Tobillo,

## 

## Table of Contents

[Table of Contents 1](#_fw3z1gxw7fik)

[Introduction 2](#_nj4vql66fib)

[Data Terminology 4](#_i6nk7zqndu8)

[Data Processing: 5](#_pcewozu9ctv6)

[Plots: 18](#_cr8evec3qnd0)

[Model Implementation](#_elh43k3khq8q) 44

[Results](#_swdmerho8ta) 47

[Resampling](#_x7zc9r35bk8s)53

[Conclusion 5](#_m13cvawutfxt)9

[Suggestions for Improvements and Future Work:](#_isy2utfjtlkf) 61

[References](#_6dx5t57807sf) 63

## 

## Introduction

***Background Information:***

The sport of Mixed Martial Arts (MMA), popularized and facilitated by the UFC, is a globally recognized combat sport that incorporates a wide variety of fighting techniques and skills that come from a variety of martial art disciplines. This is a full contact sport that allows athletes to compete using striking and grappling techniques both while standing and on the ground. There is a unified set of rules followed by all MMA competitions, this sport attracts fighters from multiple different disciplines and creates a unique and exciting blend of athleticism and energy.

In UFC events, the fighters are assigned to either the blue or red corner. The red corner is meant for the higher-ranked fighter who enters the ring second, whereas the blue corner is kept for the lower- ranked fighter. These distinctions have the potential to bring on psychological implications, since the higher-ranked fighter is presumed to have a much more competitive edge.

The appeal of MMA lies in its unpredictability and the different sets of fighting skills that each fighter competes with. However, the availability of detailed fight statistics and outcomes offer a unique opportunity to analyze the sport in a quantitative manner. By examining certain metrics, such as strikes, past performance, defense rates, etc., our data analysis can help uncover patterns and trends that can contribute to a fighter’s success.

***Purpose and Motivation:***

The purpose of this project is to develop a predictive model that will analyze fight statistics to determine the outcomes of UFC matches. By using past MMA data, this project is tasked with identifying key performance metrics involved in a fighter’s success in this sport. Not only will the insights that we derive from this analysis contribute to understanding the dynamics of MMA, but it could also provide a framework for predicting outcomes in other sports.

The motivation for this project comes from the increased demands in sports analytics, especially in using data to gain competitive advantage. MMA has a very multifaceted nature, making it a unique sport where success outcomes are dependent on a combination of physical abilities, strategies, and adaptability. Analyzing a sport with the complex nature that MMA has, provides a challenging and rewarding opportunity to apply different data science techniques.

***Objective:***

The objective of this project is to design and implement a predictive model that utilizes fight statistics to forecast the outcomes of UFC matches. To guarantee accuracy and fairness, the model incorporates resampling techniques to address data imbalances, particularly between different match outcomes.

The main goal of this project is to determine whether the fighter in the red corner, typically assigned based on a higher win record, has a statistically higher likelihood of victory compared to their opponent who is assigned to the blue corner. By analyzing different comprehensive fight statistics (strike accuracy, grappling success rates, defense effectiveness, and previous match outcomes), this project aims to uncover key performance indicators that have a significant influence on the outcome of a fight.

In addition to predicting match results, this model will provide useful insights into the competitive sport of MMA, drawing attention to the different trends and the relative importance of different fight metrics. Ultimately, the project aspires to further enhance the understanding of the sport, and uncover the actionable insights for use by athletes, coaches, and analysts.

## Data Terminology

Methods of Victory

* Submission (SUB): Physical or verbal tap out
* Knockout (KO): Athlete is knocked unconscious
* Technical Knockout (TKO): The referee stops the contest due to strikes or impact
* Judges' Decision (U-Dec, S-Dec): Fight is not finished by a submission, KO, or TKO, and judges deliver their scorecards for a final decision

Terms to Know

* Unanimous: All judges pick the same athlete as the winner
* Split: One judge picks one athlete. The other two judges pick the other athlete
* Majority: Two judges pick the same athlete as the winner but the final judge says fight was a draw
* Draw: Unanimous, majority or split

Data Shorthands

* Str: Strike
* Sig: Significant
* Sub: Submission
* Pct: Percent
* Avg: Average
* Dif: Difference
* TD: Take-down
* Att: Attempt

## Data Processing:

***Python Libraries used:***



*(If not stated here, other libraries are in code below)*

***Data Cleaning:***

The data cleaning process was a multi-step approach designed to address structural inconsistencies, missing values, irrelevant features, and excessive zeros, ensuring the dataset was ready for accurate analysis and modeling.

The first step in the process involved handling the temporal information contained in the Date column. Initially stored as strings, these data values were converted to a datetime format using Python’s pd.to\_datetime() function. This transformation enabled accurate handling of temporal data, such as sorting and filtering by specific time periods. Any invalid or improperly formatted dates were coerced into NaT (Not a Time) values, ensuring the dataset remained consistent and free of erroneous entries. Following this, a time restriction was applied to include only fights occurring between 2023 and 2024. This restriction was implemented to maintain the relevance of the data, aligning it with current trends and ensuring it reflected recent activity in UFC fights. By excluding outdated records, the dataset was narrowed to a manageable and meaningful timeframe, which also reduced potential noise in subsequent analyses.

The dataset contained multiple categorical columns, such as Location, EventName, and fighter names. While these columns might provide context in specific use cases, they were deemed irrelevant for this analysis, as they lacked direct correlations with the fight outcomes. Including these columns could have introduced noise into predictive models, complicating the analysis without adding substantial value. To streamline the dataset, all categorical columns except for Winner were removed. The Winner column, which directly indicates the outcome of each fight, was critical for modeling. To make this column compatible with machine learning algorithms, its values were transformed into a binary numerical format: "Red" was replaced with 1, and "Blue" was replaced with 0. This transformation standardized the representation of fight outcomes, enabling efficient computation and improving model interpretability.

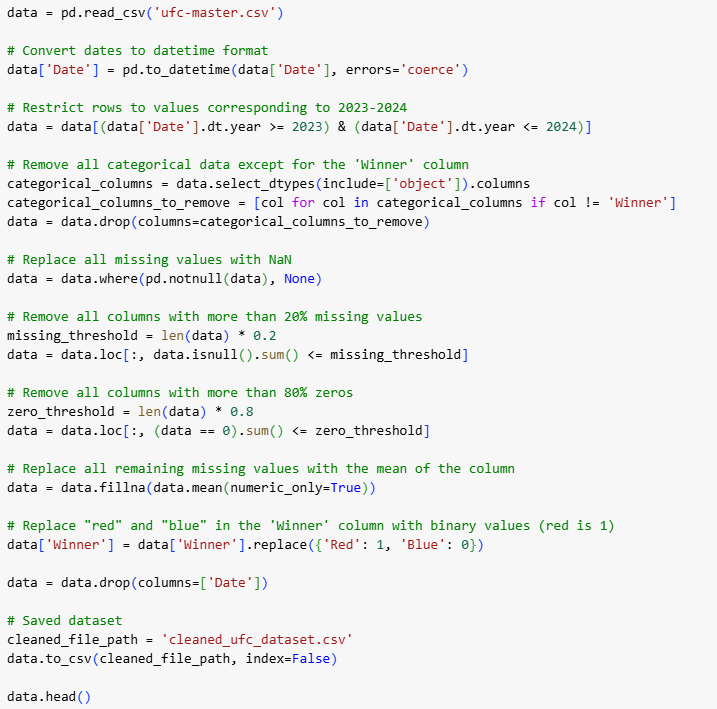
Handling missing data was a pivotal step in the cleaning process. Columns with excessive missing values were identified and removed based on a threshold of 20%. Any column where more than 20% of the entries were missing was excluded to prevent distortions that could arise from sparse data. This threshold balanced the need to retain as much information as possible while ensuring data quality. For the remaining columns, missing values were replaced with the mean of their respective columns. Mean imputation is a widely used technique that preserves the overall distribution of numerical data, minimizing the risk of introducing biases. By addressing missing data systematically, the dataset was made complete and consistent, ensuring it was well-prepared for subsequent analysis and modeling tasks.

In addition to missing values, the dataset was examined for columns with an unusually high proportion of zero values. Columns where over 80% of the entries were zeros were deemed uninformative and removed from the dataset. Such columns lack variability and are unlikely to contribute meaningful insights or predictive power to analytical models. Excessive zeros can distort model training and lead to overfitting or poor performance. By identifying and removing these columns, the dataset was streamlined to retain only the most informative and relevant features. This step further optimized the dataset for analysis by eliminating redundant or irrelevant data.

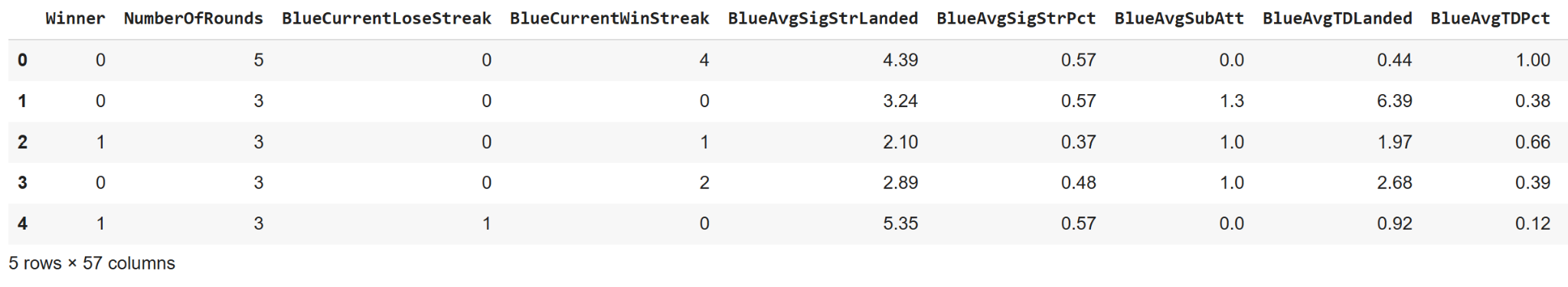
After addressing missing values and excessive zeros, minor adjustments were made to finalize the dataset. The Date column, which had already served its purpose in filtering the data to the specified timeframe, was dropped. Retaining this column could introduce unintended temporal biases during model training and analysis, so its removal was necessary. The cleaned dataset was then saved as a new file, cleaned\_ufc\_dataset.csv, to ensure reproducibility and facilitate downstream tasks. This final step marked the culmination of the cleaning process, resulting in a dataset that was concise, consistent, and optimized for further analysis.

Through these steps, the data cleaning process successfully transformed the raw dataset into a well-structured, high-quality dataset ready for predictive modeling and analysis. Each step was carefully executed to address specific challenges while retaining the integrity and relevance of the data.

***Code for the data cleaning shown :***



***Cleaned Data Set***



*(Output is truncated)*

The table includes key features such as the binary Winner column, indicating fight outcomes (0 for Blue, 1 for Red), alongside numerical features like NumberOfRounds, BlueCurrentLoseStreak, BlueCurrentWinStreak, and various performance metrics such as BlueAvgSigStrLanded (average significant strikes landed by the blue corner), BlueAvgSigStrPct (percentage of significant strikes), and BlueAvgTDLanded (average takedowns landed). The cleaned dataset has been streamlined to focus on relevant numerical features and is free of irrelevant or sparse data, ensuring it is ready for analysis and modeling.

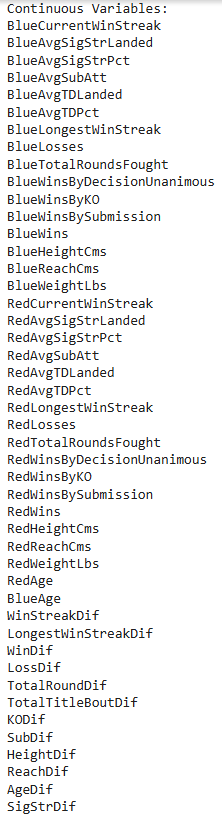
***Feature Selection:***

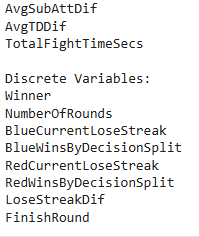
Following the data cleaning process, we then moved on to selecting which predictive variables would be best at contributing towards high accuracy classification models. We went through various techniques when selecting which features we wanted to use in our final few models, this is due to our initial cleaned data having over fifty columns in it, each being its own feature.

The first thing we looked at when trying to find proper features was the type of data that was available to us, thus, we separated the variables into two groups, continuous variables, and discrete variables. The way we defined a continuous variable, was a variable that had ten or more unique values, and a discrete variable was one with less than ten. This resulted in us having six variables with less than ten unique values, deeming them to be discrete. We then proceeded to get rid of all the discrete variables, because they have very little variation and prediction power. This will benefit us more in the future since removing them can reduce noise, and prevent overfitting in our models.

***Code for separating continuous and discrete variables:***



***Output:***





***Continuous variables in a python list (above)***

Furthuring the feature selection process, we moved on to using statistical testing to

determine the relationship between each feature and the target variable (Winner). This, in turn, will help see which features have the highest predictive power in a model. The statistical test used was a basic T-test, which was run for each individual feature. For each of these tests, we were interested in looking at the p-values, and set a threshold of having a p-value of less than 0.05 to be considered a statistically significant feature. Features with p-values higher than 0.05 were considered to be weakly related to the target variable, and would only cause more problems if we were to leave them in a predictive model, thus we removed all of them. After keeping all of the statistically significant features, we were left with thirty predictor variables, which we still saw as too much for running multiple classification models.

***Code for obtaining p-values:***

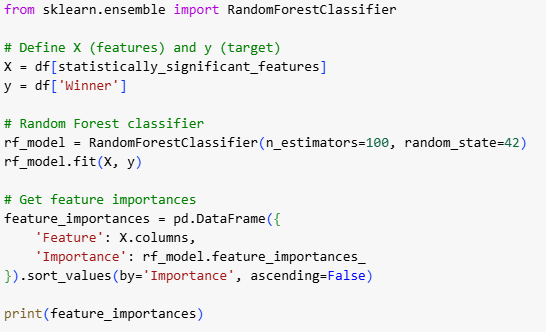


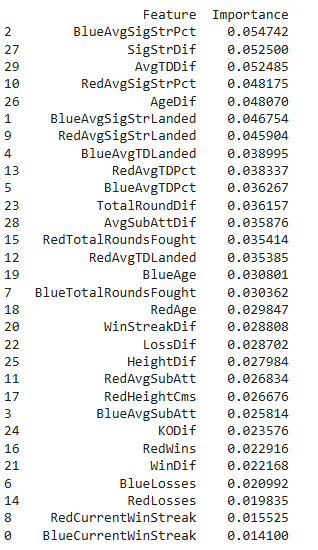
***P- Values Output:***

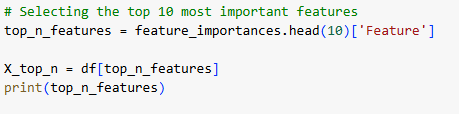
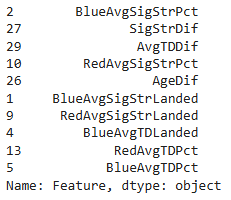


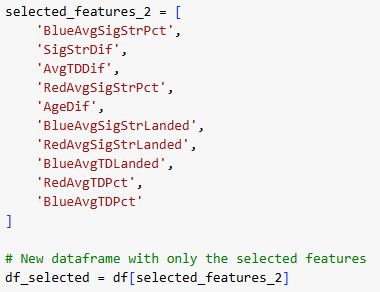


Getting deeper in the feature selection process, we moved on to using the Random Forest Algorithm as a feature evaluator. Random Forest is a powerful tool in data science, which is used plenty on creating predictive models, but it can also be used in the feature selection process. What Random Forest does is find the importance of each feature by using ensemble techniques that build multiple decision trees trained on bootstrapped data samples. It then looks at each of these trees and sees what features help decrease the impurity of said tree. A feature that consistently splits the data well, leads to a tree with a higher purity level, signifying that the feature will have a higher importance. Features with lower importance scores were removed from our dataset since they do not significantly affect our model performance.

***Code for Random Forest Classifier:***

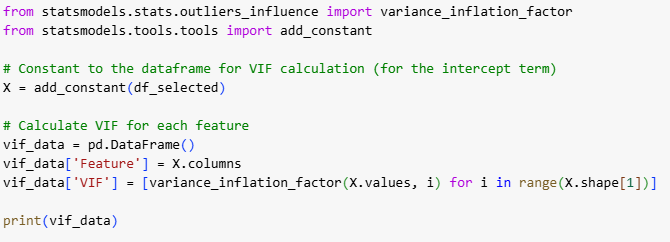
***Output:***

After taking a look at the importance of all of our features, we decided to keep the ten features with the highest importance which are listed below with the code;

Putting the top ten predictors into a pandas dataframe:

With only ten predictors remaining, we moved on to check and see if there was any type of relationship between any of these variables. We checked for these correlations by looking at the Variance Inflation Factor (VIF), which is a popular statistical measure that helps identify and quantify multicollinearity in data. The reason we want to check for this is so the models we create do not have too much overlapping information, this can lead to the models being severely affected by any slight change in the data. Multicollinearity can also lead to an inflated standard error, which makes it harder to determine whether a model has high predictive power or not. The way we read VIF scores is that the higher the score, the higher correlation a variable has with the others. The VIF scores for each individual feature and a constant term are listed below.

***Code for VIF:***



## 

## ***VIF Output:***

## 

## 

This image shows a list of the ten selected features and their correlated Variance Inflation Factor (VIF) scores, the scores are shown on the right, with inf meaning infinite or a really high value.

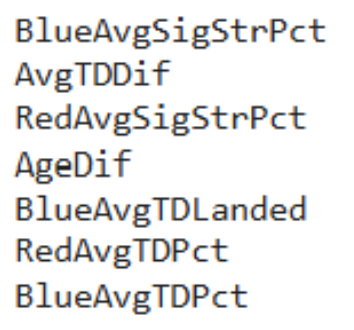
After running the VIF score, it was determined that the best course of action was to remove all of the predictor variables with high scores. In the image above it is clear that other than the constant term, three of our variables have VIF scores that are too high for a sustainable model. Thus, we removed these three variables from our final dataset, and we were left with seven remaining features. After Implementing our various different feature selection tests, we were left with the following predictors: 

Image displaying the 7 features that were left after feature selection, these features will be the ones we use in training our classification models.

## After obtaining the final predictors, plots were made to obtain a better insight into the predictors that would be used for the model.

## 

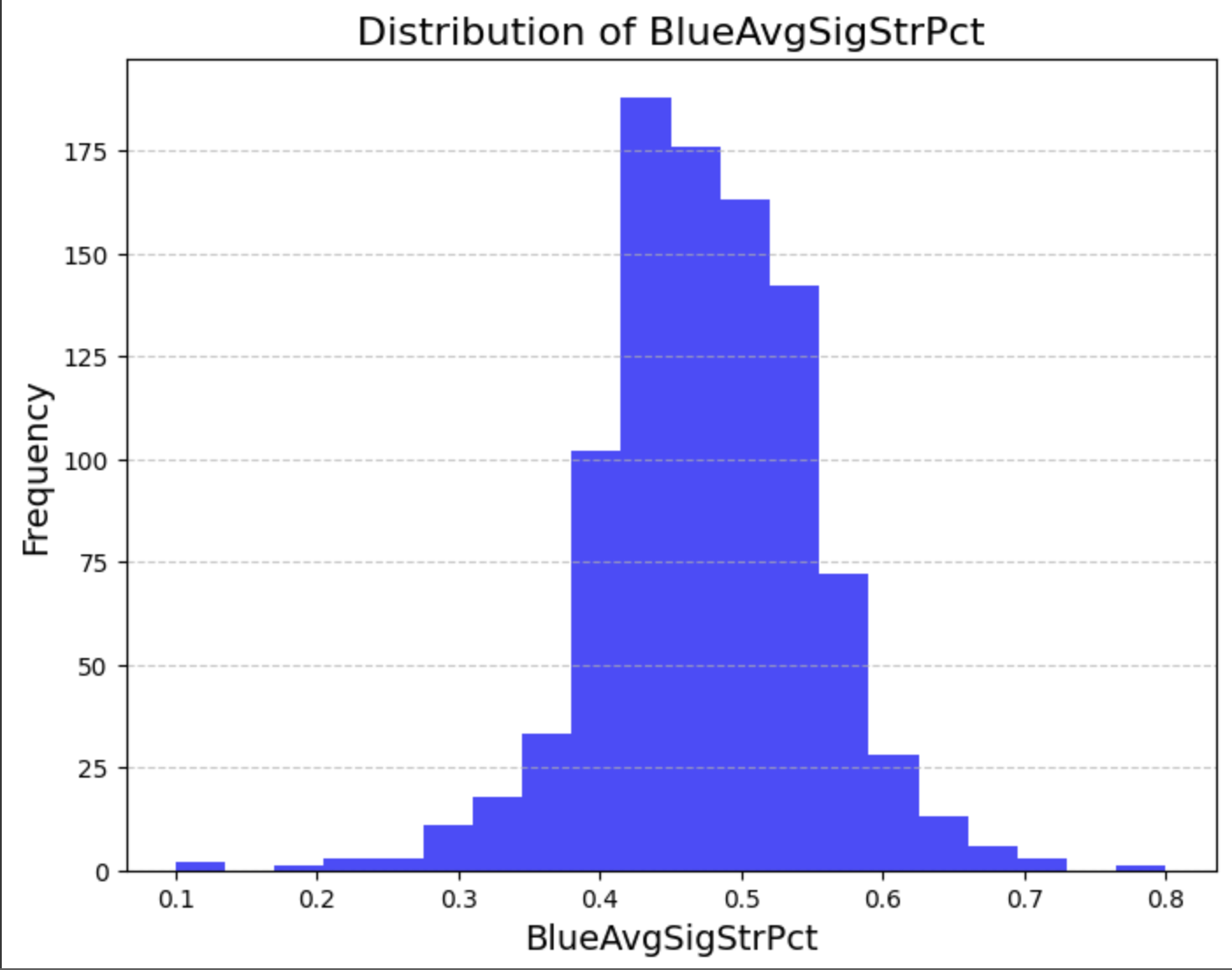
## 

## 

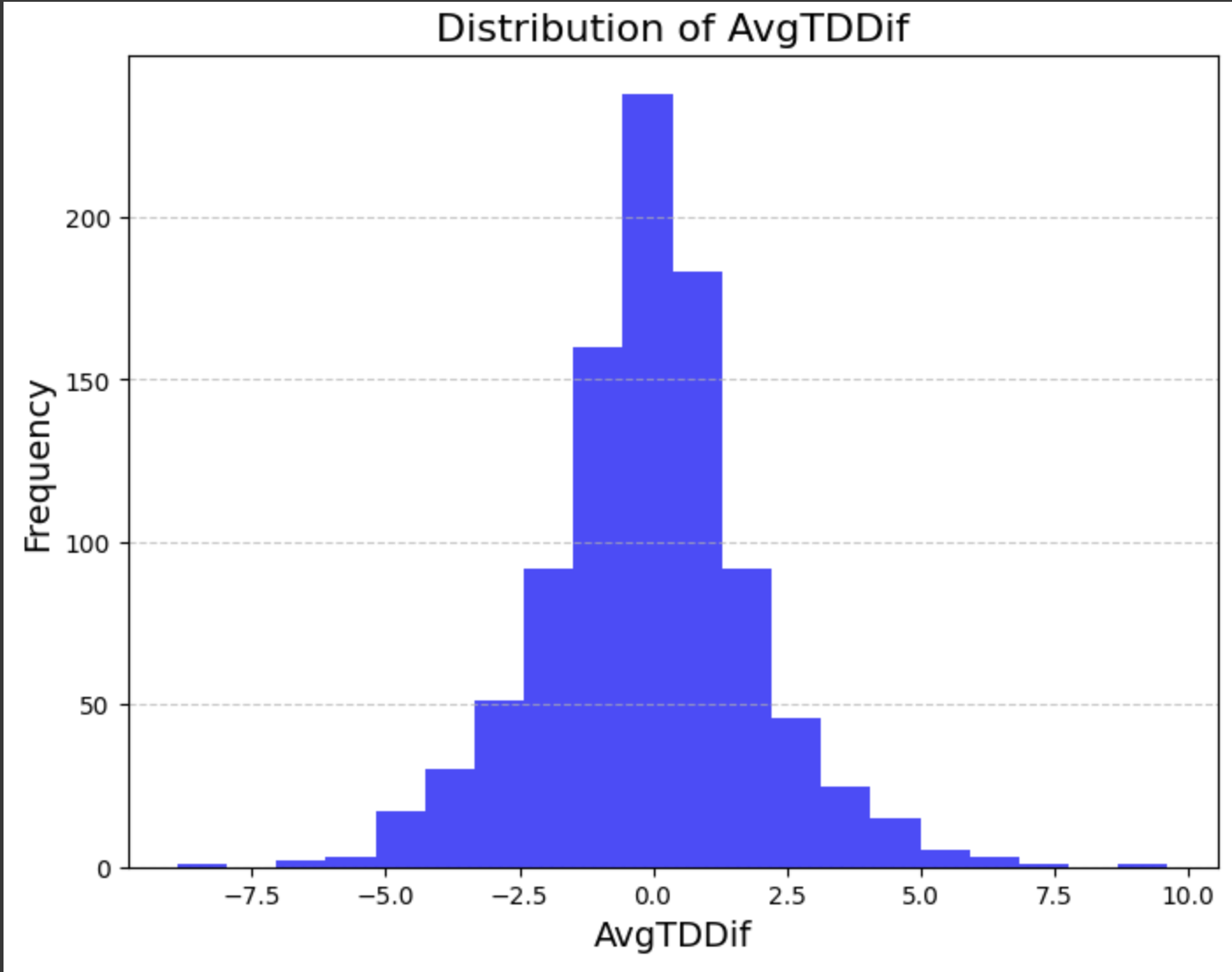
## Plots:

***Python Libraries used for plots:***

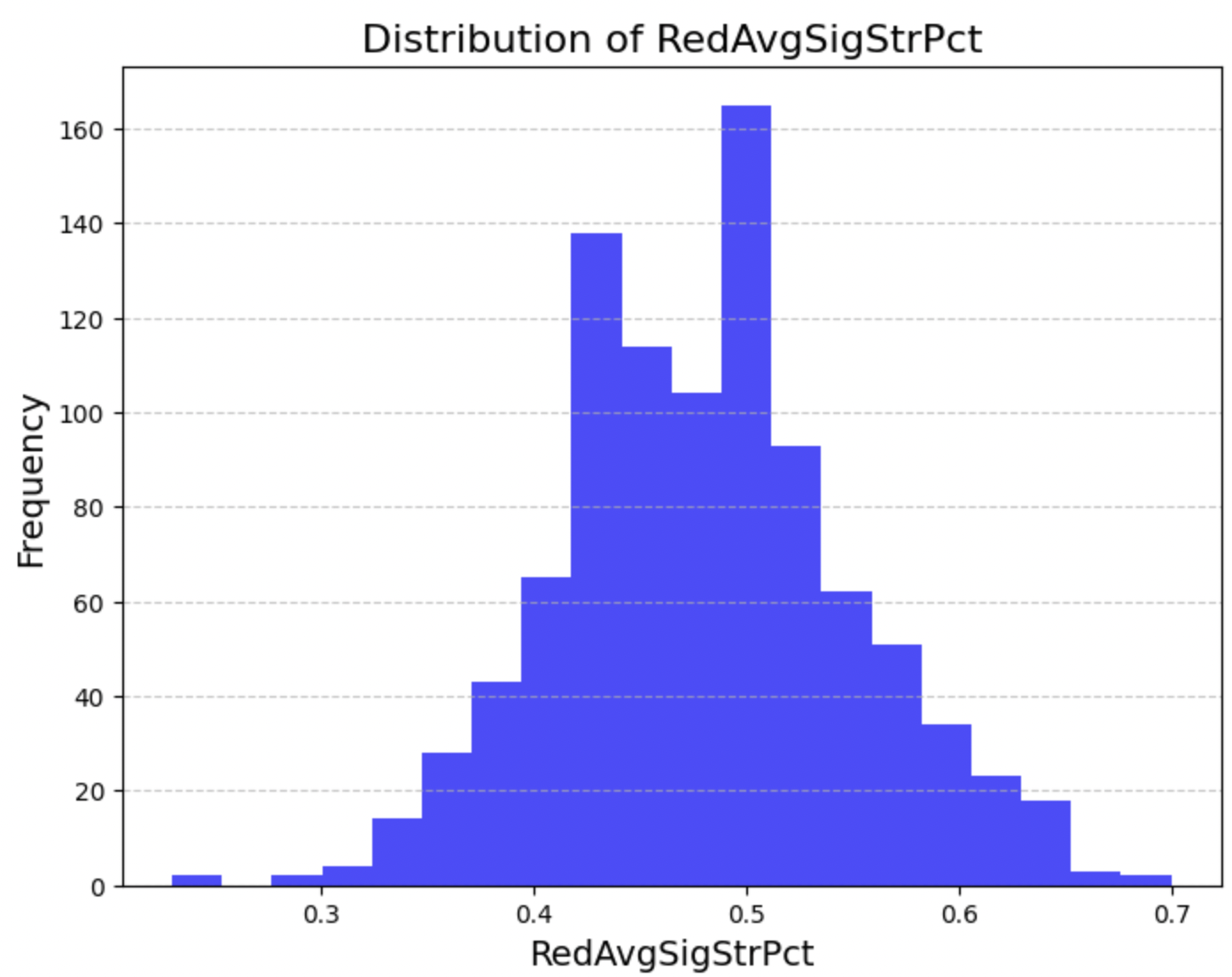


***Histograms:*** 

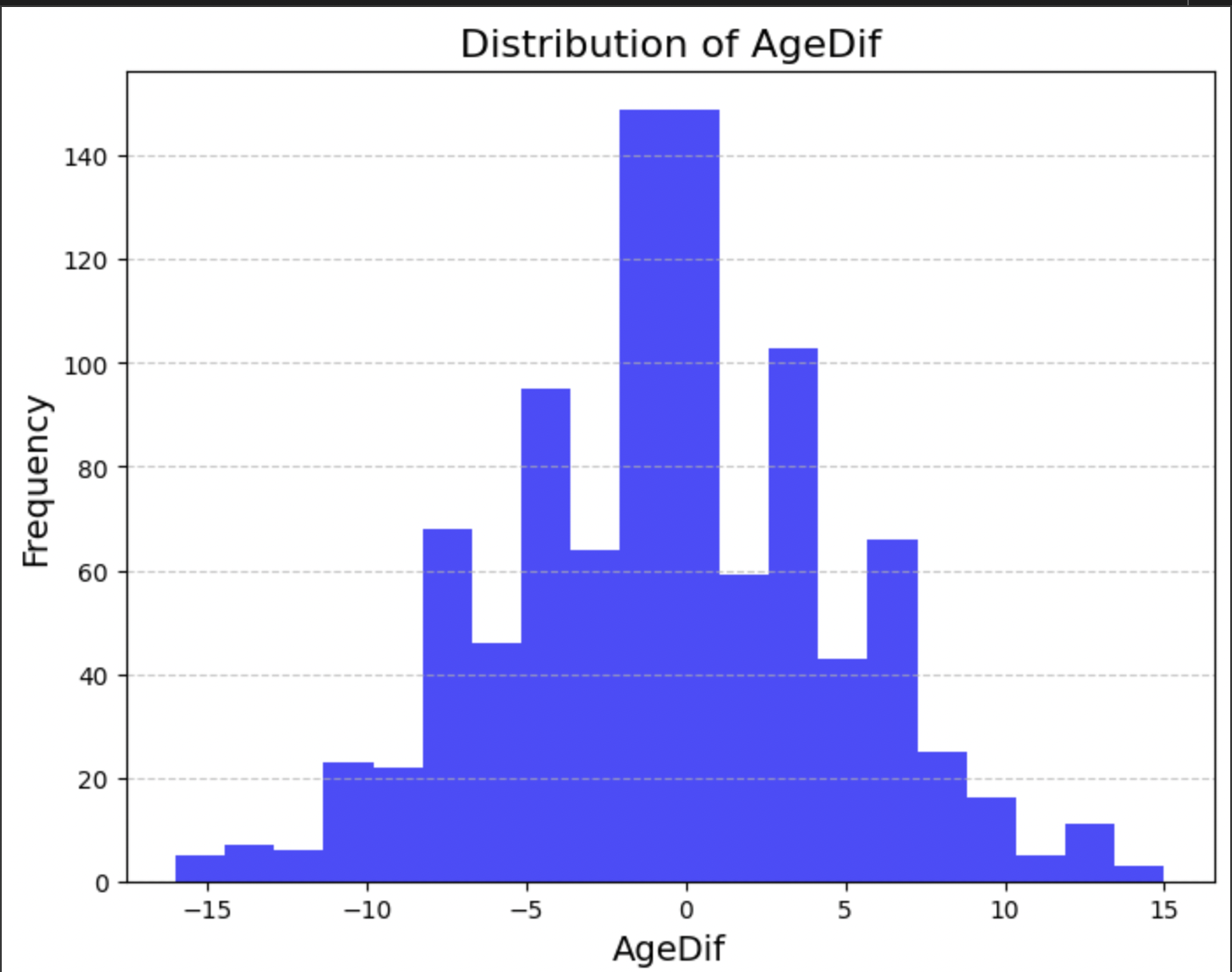
The distribution is slightly skewed to the right, with most variables between 0.4 and 0.6. This measures the significant strike percentage for the fighter in the blue corner. From this histogram it can be seen that most fighters land 40-60% of their significant strikes, with very few having a percentage below 30% or above 70%.

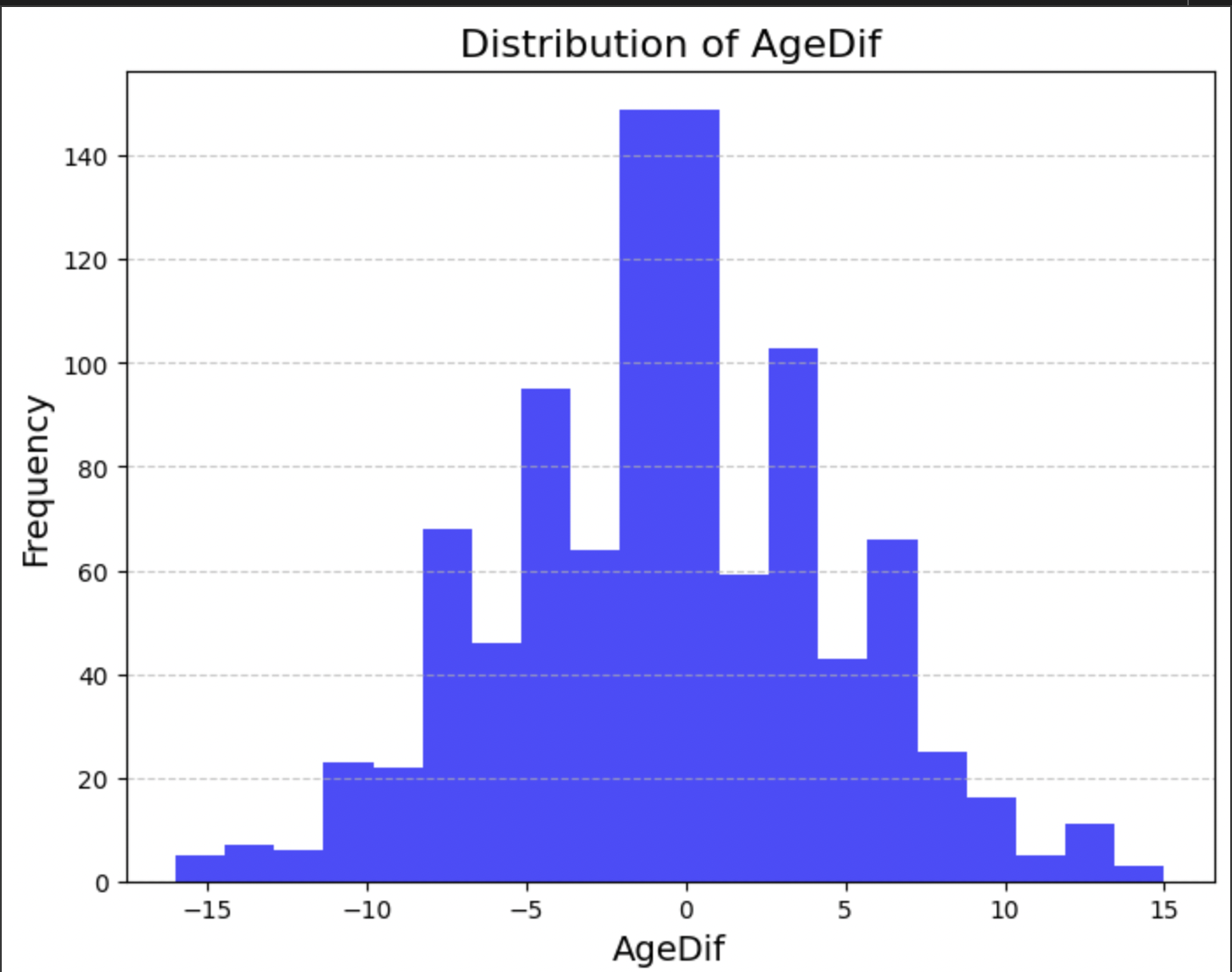


This histogram represents the difference in takedown defenses between fighters. There is a normal distribution indicating that most fights have evenly matched fighters in terms of takedown defense.

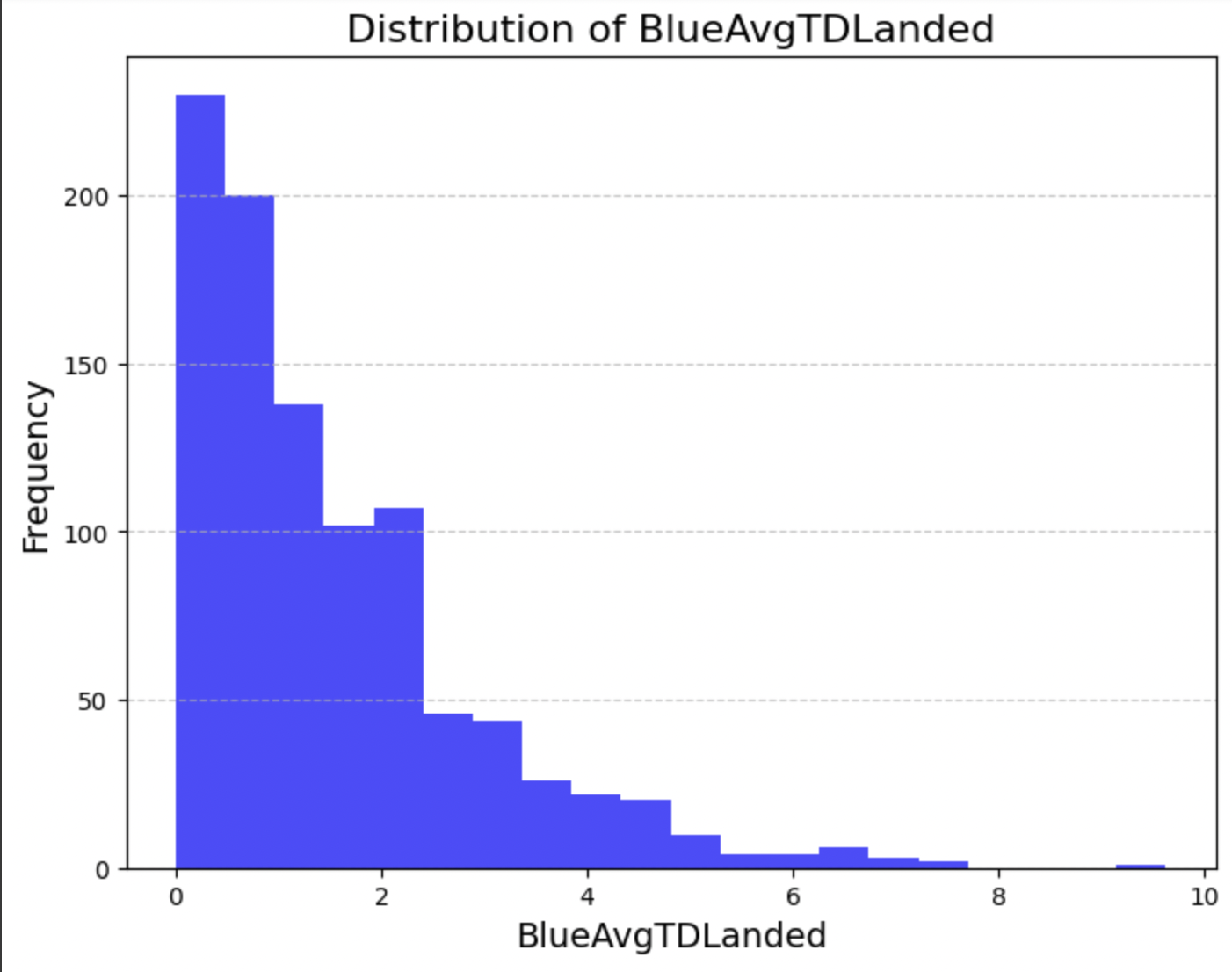


This histogram represents the significant strike percentage for fighters in the red corner. It is a bimodal distribution. The significant strike percentage for fighters in the red corner clusters into two separate groups. This may be attributed to the possibility that fighters have specific styles or levels of accuracy.

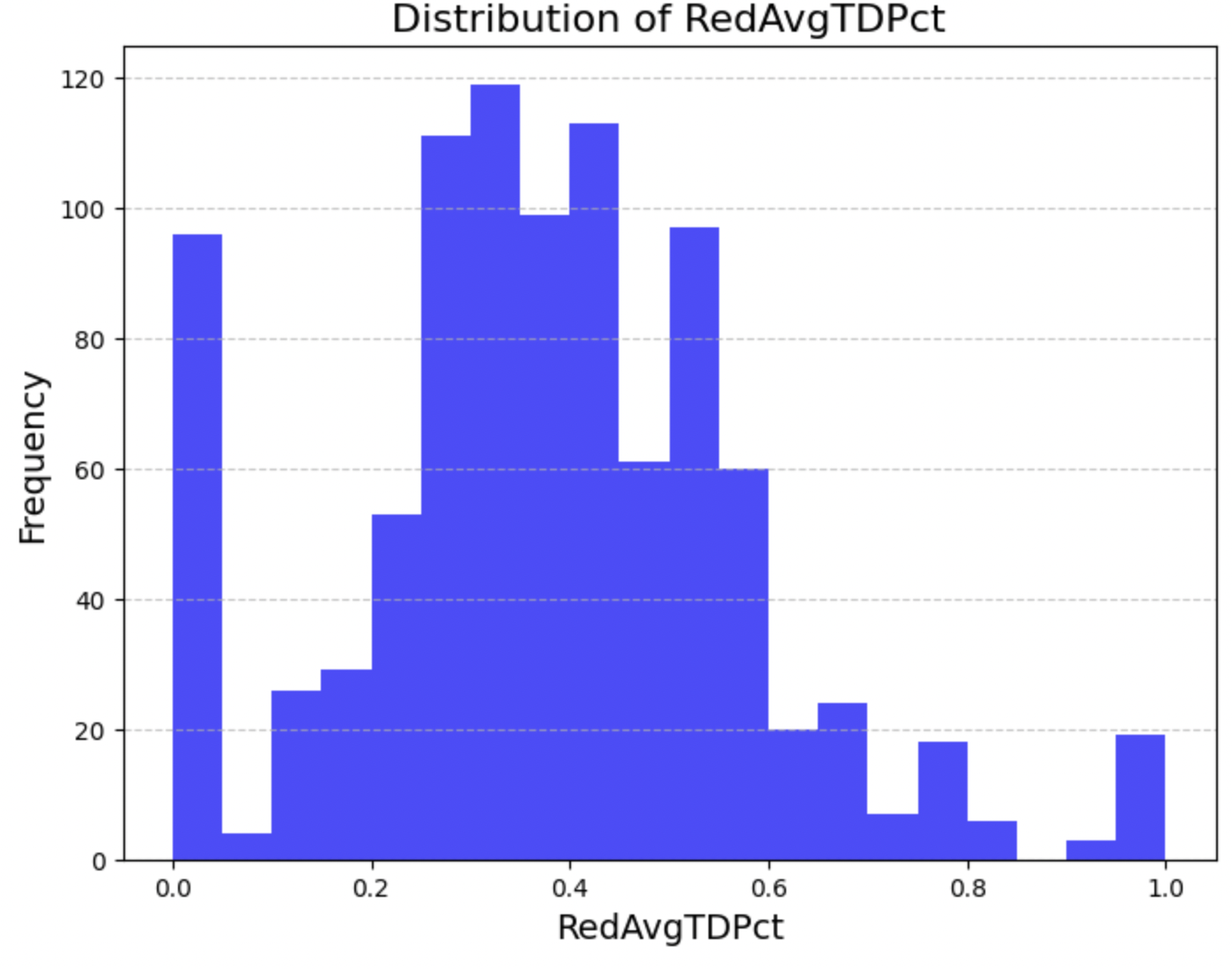




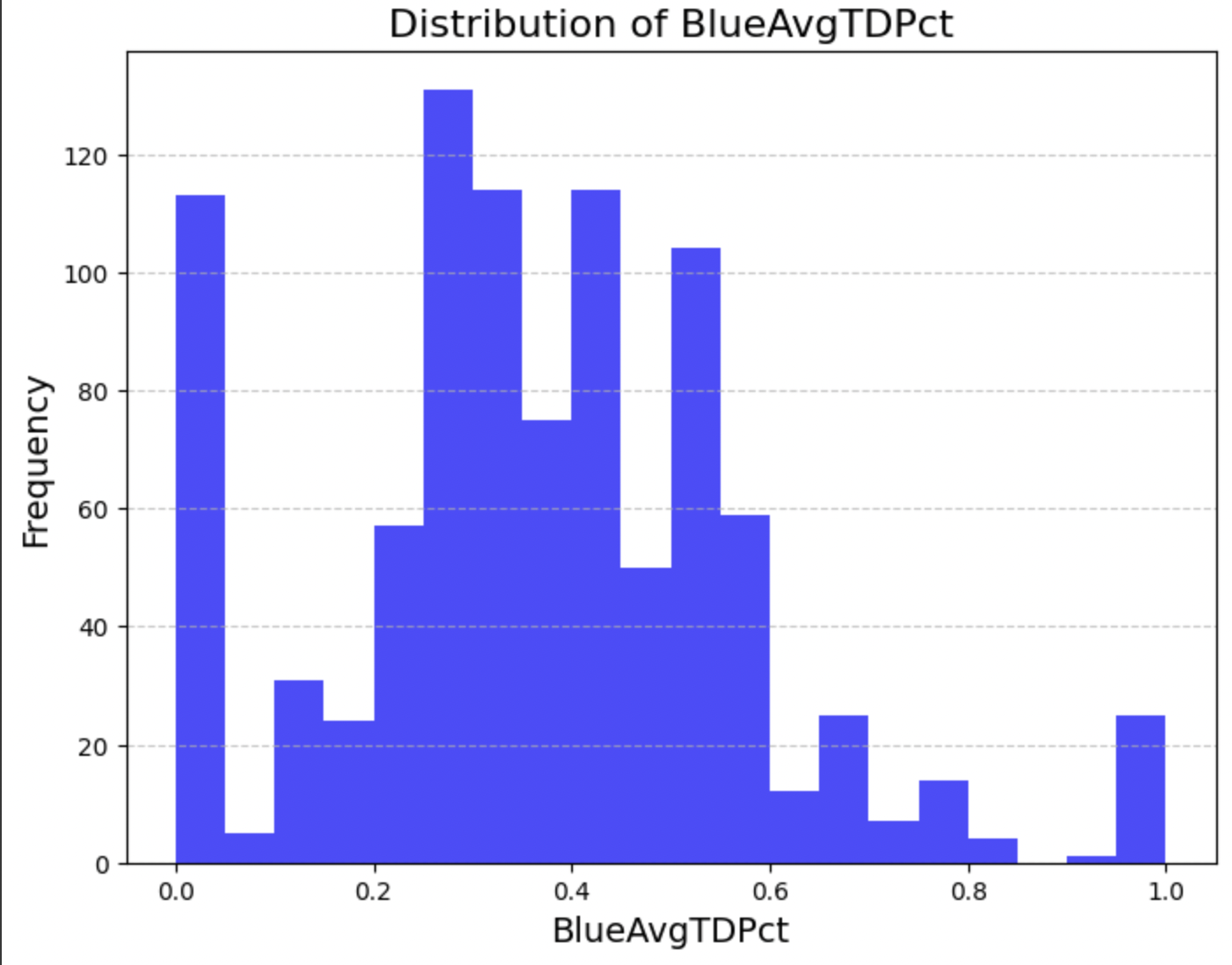
AgeDif measures the age difference between fighters, the histogram shows an approximately normal distribution. This indicates that most fights involve opponents in a similar age group. There are very few fights involving significantly younger or older fighters compared to their opponents.



BlueAvgTDLanded measures the average number of takedowns landed by the fighter in the blue corner. This distribution is right skewed, most of the values are concentrated between 0 and 2. According to the histogram, most fighters in the blue corner land very few takedowns per fight. Very few manage to land more than 4 takedowns.



RedAvgTDPct measures the takedown percentage for the red corner. The distribution has a wide range but shows clustering around 0.4 and 0.6. Fighters in the red corner usually perform around these values, but there are some outliers that show very high or very low takedown percentages.

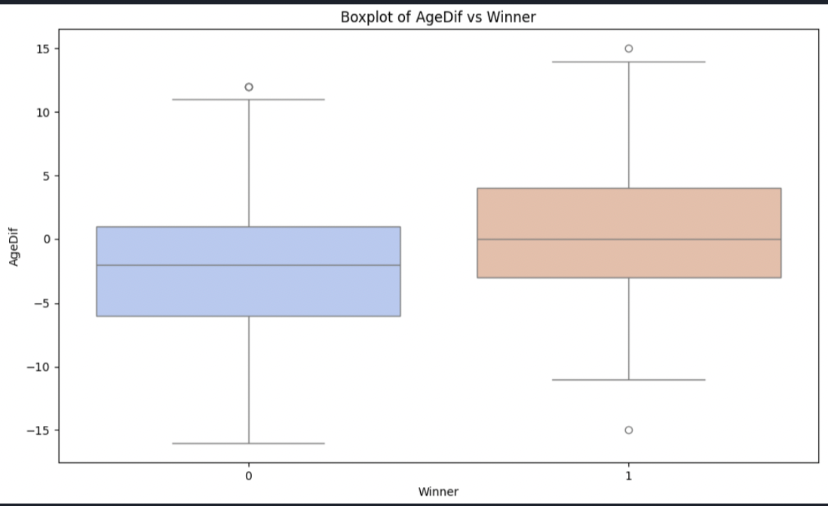


BlueAvgTDPct represents the takedown percentage for fighters in the blue corner. Similar to the takedown percentage for fighters in the red corner, this histogram also shows a lot of variability, but there are distinct clusters around 0.2 and 0.4.

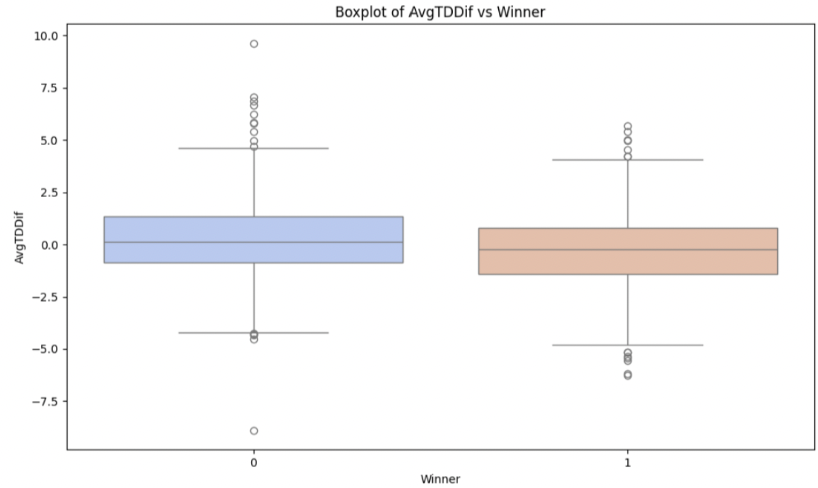
***Box Plots (0 is loss, 1 is win):***



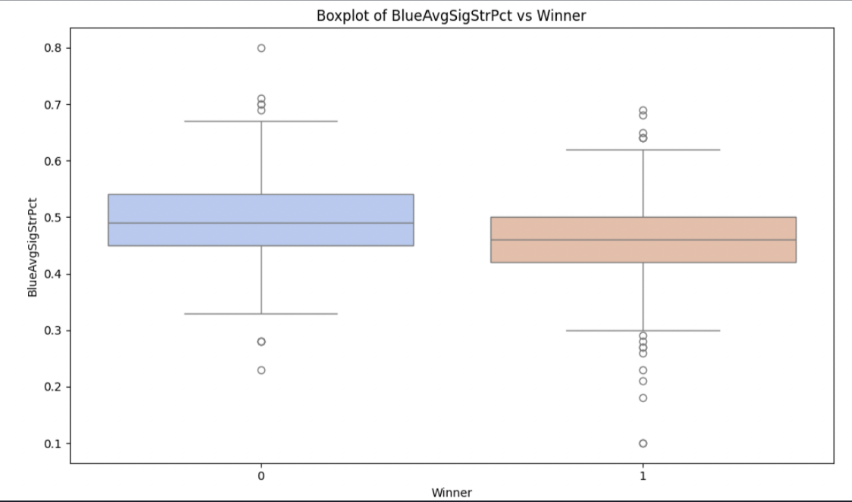
This box plot compares the RedAvgSigStrPct (average significant strike percentage for red corner fighters). The median significant strike percentage is higher for winners, indicating that a higher RedAvgSigStrPct correlates with winning. The spread of values is similar in both groups, but winners have less outliers with lower strike percentages.



This box plot compares the AgeDif (age difference) between fighters based on fight outcomes. Winners have a higher median age difference, skewing slightly older than their opponents. The spread of age differences for winners is wider, and there are notable outliers in both groups. This shows that older fighters may have a small advantage in winning.

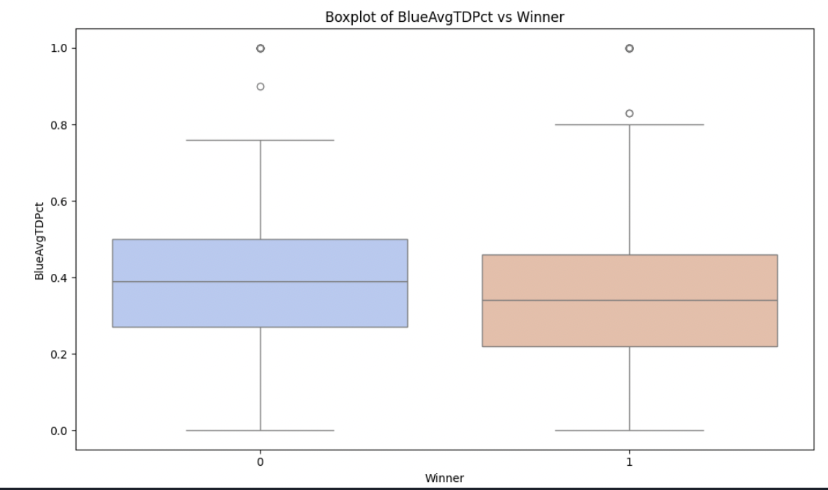


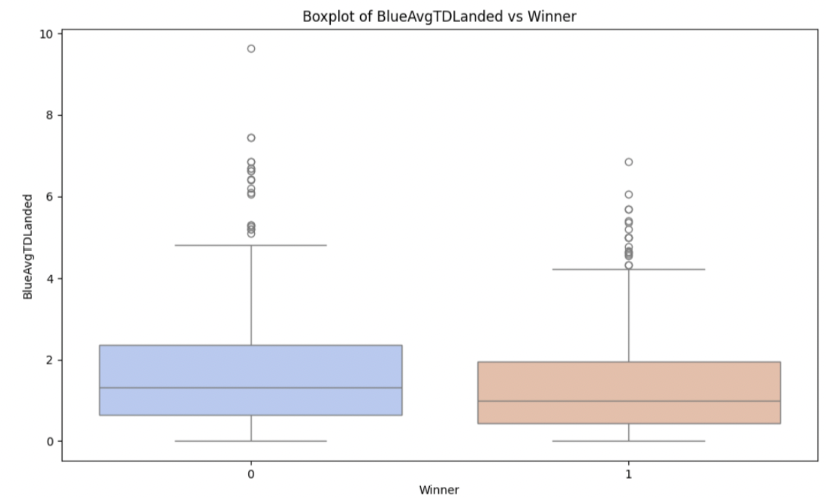
This box plot compares AvgTDDif (average takedown difference). Interestingly, the median AvgTDDif for winners is slightly lower than for losers, potentially suggesting that takedown differences alone may not be a decisive factor in winning.



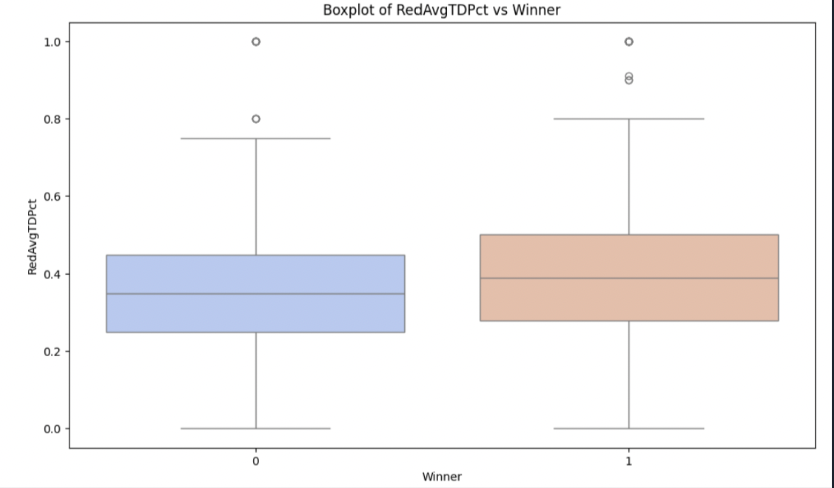
This box plot shows the BlueAvgSigStrPct (average significant strike percentage for blue corner fighters). Once again, the median strike percentage for winners is slightly lower than for losers, suggesting that higher striking accuracy on its own might not guarantee a win. However, the interquartile range for winners is tighter, showing there is more consistent striking performance.

This box plot compares BlueAvgTDPct (average takedown percentage for blue corner fighters). The medians for both groups appear similar, indicating that takedown success rates might not play a big role in differentiating the winners from losers. However, the interquartile range for winners is wider, suggesting that there is more variability in takedown success with winning fighters.



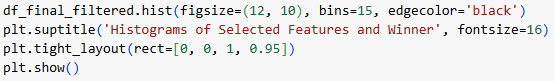


This box plot compares BlueAvgTDLanded (average takedowns landed for blue corner fighters) The median number of takedowns landed is higher for the losers, suggesting that landing more takedowns does not necessarily guarantee a victory. Losers also exhibit a wider interquartile range, showing more variability in takedown performance. Winners, on the other hand, show a narrower IQR, meaning more consistent performance.



This box plot compares RedAvgTDPct (average takedown percentage for red corner fighters). The median for winners is slightly higher than for losers, indicating that higher takedown success rates may contribute to winning. However, the interquartile ranges are similar for both groups, suggesting comparable variability in takedown percentages. Winners display slightly more consistent performance overall.

***Code for histograms:***



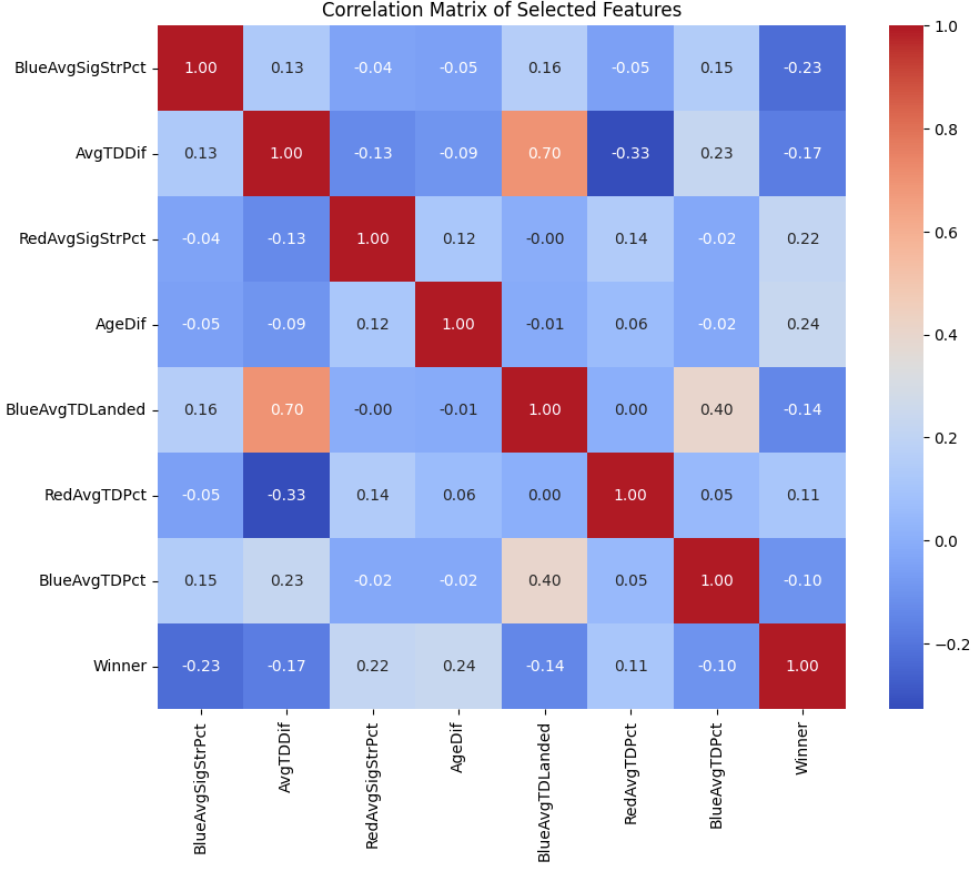
***Code for box plots:***



***Pairplot:***



This plot shows that AvgTDDif and BlueAvgTDLanded have the strongest linear relationship. Pairplots show that if one color dominates in specific regions, it suggests a strong relationship with the outcome. However, within our pairplot both red and blue winners are somewhat evenly matched within each variable with neither color being easier to predict.



***Heat map:***

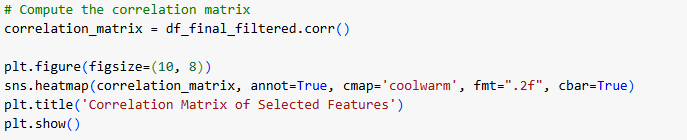
Heatmaps show the correlation between variables based on the color intensity of each cell, brighter colors means a stronger positive correlation between two variables, while darker colors mean a stronger negative correlation. Based on this knowledge it can be seen that the Winner variable has no clear correlation with other variables(also known as predictors). The only

variables with notable correlation are ones that we would expect to be highly correlated, and thus do not provide much insight, such as BlueAvgTDLanded and AvgTDDif. Overall the data has very little correlation and does not give any insight to predicting whether blue and red fighters are more likely to win.

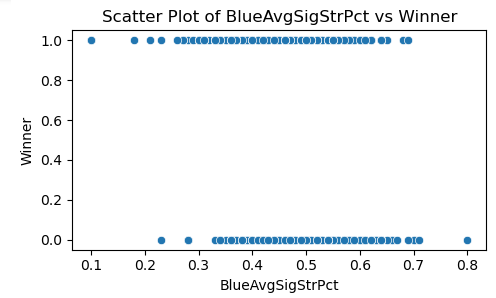
***Code for pairplot:***



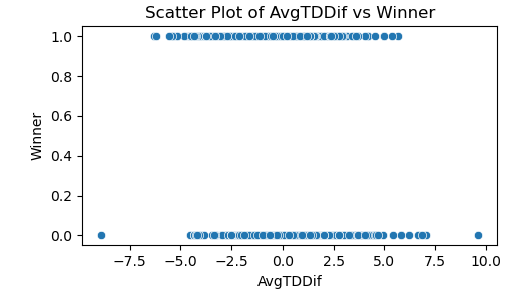
***Code for correlation matrix:***



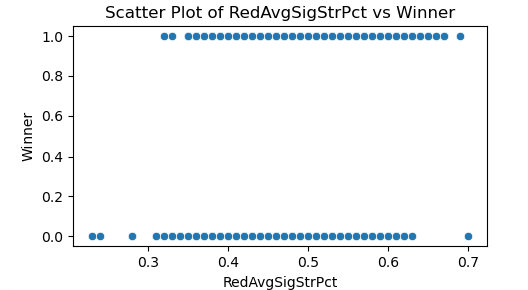
***Scatter plots:***



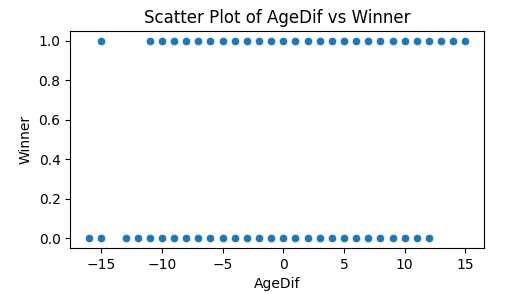
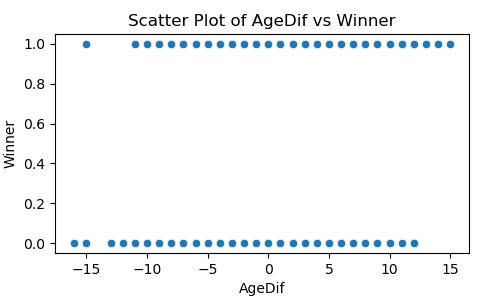
This graph indicates that at higher values of BlueAvgSigStrPct, most points are concentrated at Y=0, indicating that when Blue lands a higher percentage of significant strikes, Blue is more likely to win. While at lower values of BlueAvgSigStrPct (0.1–0.3), the data appears more evenly distributed between Y=1 (Red is the winner) and Y=0 (Blue winner), suggesting less predictive power in this range. Overall there seems to be a weak trend that higher BlueAvgSigStrPct values are associated with Blue winning, while lower values show a less clear distinction. However this is expected since the BlueAvgSigStrPct variable is more biased towards blue.



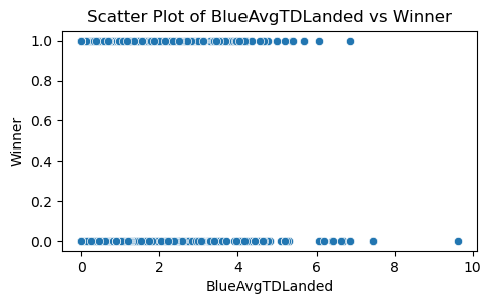
The scatter plot highlights that AvgTDDif has a noticeable influence on the fight outcome. Blue generally performs better with a takedown advantage (positive AvgTDDif), while Red thrives when they hold the takedown advantage (negative AvgTDDif). However, the overlap in values shows that this feature should be complemented with others for robust predictions.



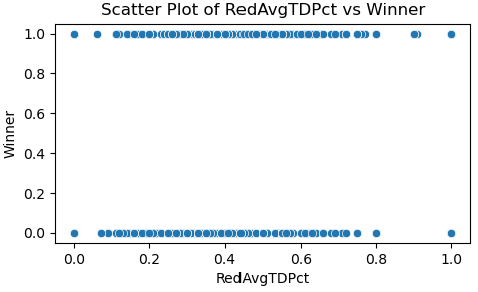
This scatter plot shows that RedAvgSigStrPct positively correlates with Red winning (Y=1). Higher percentages increase the likelihood of Red’s victory, while lower percentages give Blue a higher chance of winning. This is expected since the variable RedAvgSigStrPct is more biased towards red. However, the overlap in the mid-range values suggests this feature works better in conjunction with others for accurate predictions.



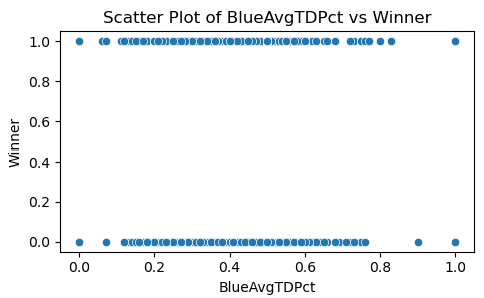
This scatter plot indicates that age difference (AgeDif) does not strongly influence fight outcomes on its own. There does not appear to be a strong or consistent trend between the age difference and the winner. Both Red and Blue win fights across the entire spectrum of age differences. The plot also shows significant overlap, particularly in the mid-range values ( -5 to +5), further suggesting that this feature does not provide strong predictive power when used alone.



This scatter plot indicates that BlueAvgTDLanded is a valuable feature for predicting fight outcomes. Higher values strongly favor Blue wins (Y=0), while lower values show mixed outcomes.These results are expected since average blue takedowns are more biased towards blue rather than red. Combining this feature with others could enhance its predictive power and provide a clearer understanding of the factors influencing fight results.

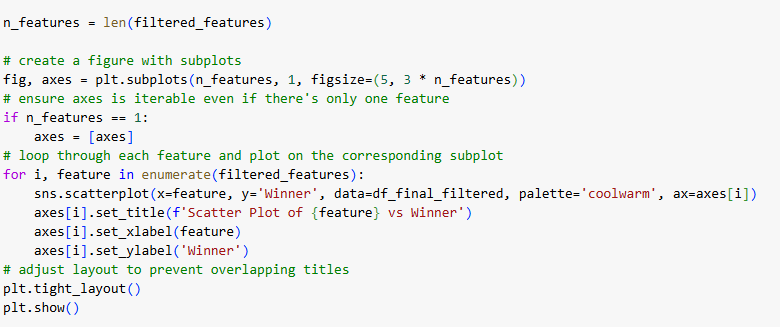


This scatter plot indicates that RedAvgTDPct is a valuable feature for predicting fight outcomes. Higher takedown success rates strongly favor Red wins (Y=1), while lower rates lead to mixed results. These results are expected since RedAvgTDPct are more biased towards red rather than blue. This feature can be a significant contributor when combined with other relevant variables in a predictive model.



This scatter plot indicates that BlueAvgTDPct is a valuable feature for predicting fight outcomes. Higher takedown success rates strongly favor Blue wins (Y=0), while lower rates lead to mixed results. These results are expected since BlueAvgTDPct are more biased towards blue rather than red. This feature can be a significant contributor when combined with other relevant variables in a predictive model.

***Code for scatter plots:***

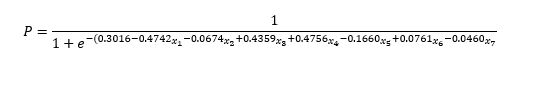


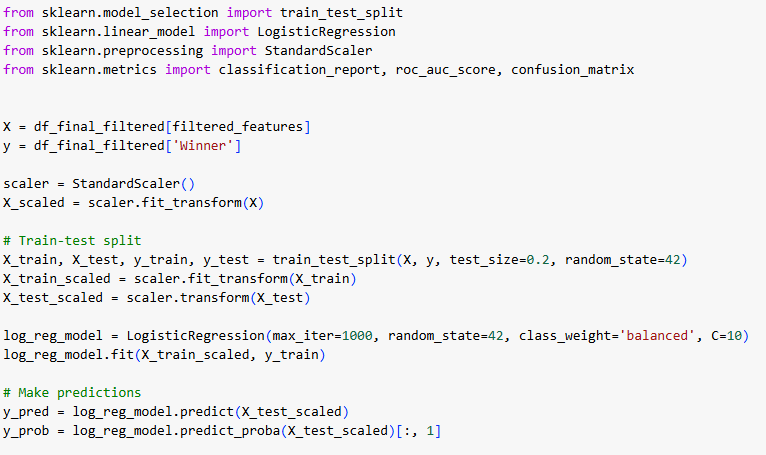
## 

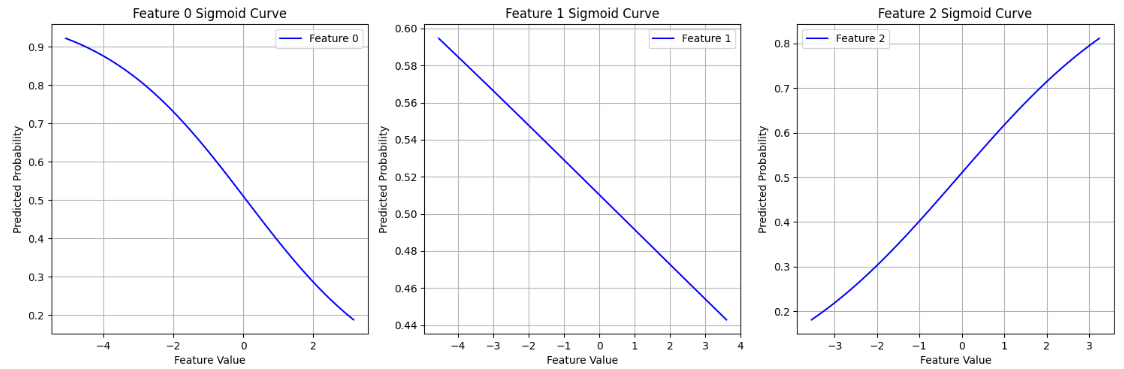
## 

## Model Implementation

For model implementation we decided to use a Logistic Regression model. The target variable is a binary target so using logistic regression is the most obvious model to implement. Decision trees and neural networks could be used but that is outside the scope of this project. To implement the model the key predictors were first scaled to prevent any imbalance to the model performance since certain features can outweigh others. Then a train test split was implemented to evaluate model performance (split was 80% training and 20% testing). Standardization was applied to both these sets with fit\_transform applying to training data only. The logistic regression was then trained using specific settings, including a regularization strength (C=10) to control model complexity, a "balanced" class weight to handle uneven data distribution, and a maximum of 1000 iterations to ensure the model converges during training. This model effectively combines the input features to estimate the likelihood of the positive outcome.







***Features are just labeled by index***

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

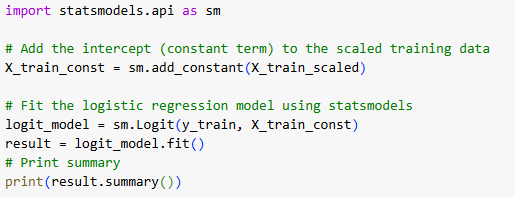
## 

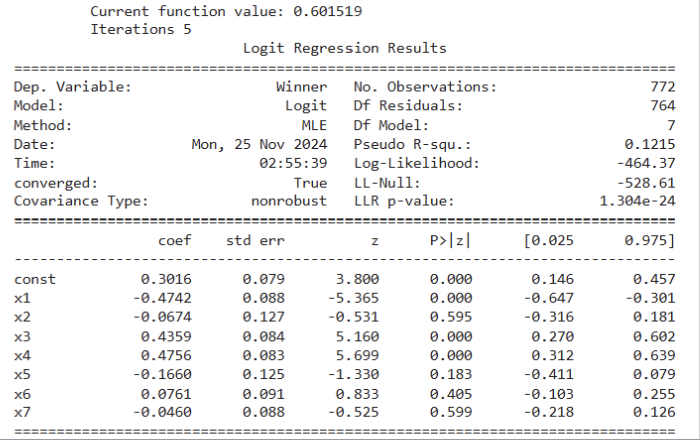
## 

## Results

This logistic regression model is designed to predict the probability of one of two possible outcomes, such as "yes" or "no," based on seven input features (x1 to x7​). Each feature is assigned a coefficient, which represents its influence on the prediction. Positive coefficients, like 0.4359, 0.4756, and 0.0761 indicate that as the corresponding feature increases, the likelihood of the positive outcome ("yes") also increases. Negative coefficients, such as −0.4742, -0.0674, 0.1660, and -0.0460 suggest that higher values of the feature decrease the probability of the positive outcome. The model uses these coefficients in a mathematical formula to compute the probability, ensuring the output is always between 0 and 1.

***Code for statistics for Logistic Regression Model:***

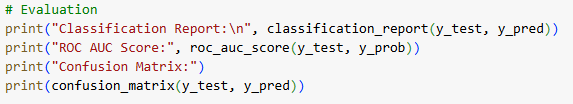


***Summary Statistics for Logistic Regression Model:***

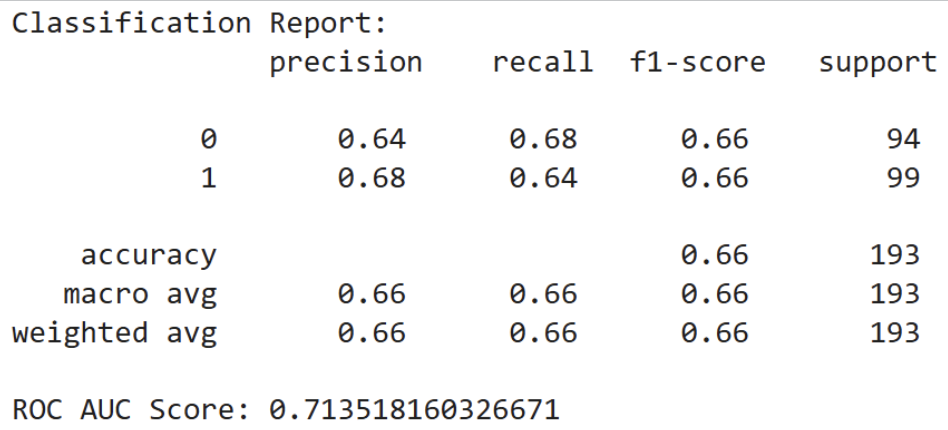
This table summarizes the results of a logistic regression model used to predict a binary outcome ("Winner: Yes or No") based on seven features (x1 to x7​) and a constant term (const). The results also include statistical details. The std err column shows the variability of each coefficient estimate, and the z-value indicates how many standard deviations the coefficient is away from zero. The P>∣z∣ column contains p-values, which test whether each feature significantly contributes to the model. A p-value below 0.05 implies a significant relationship. For instance, x1 (p=0.000) and x3 (p=0.000) are statistically significant predictors, meaning they strongly influence the outcome. In contrast, features like x2 (p=0.595) and x7 (p=0.599) are not statistically significant, indicating that their contribution to predicting the outcome is negligible in this model. The confidence intervals ([0.025, 0.975]) provide a range of likely values for each coefficient, offering additional context for their reliability.

The overall model performance metrics also provide insights. The pseudo R-squared value (0.1215) suggests the model explains around 12.15% of the variability in the outcome, which is modest for a logistic regression model. The log-likelihood (-464.37) indicates how well the model fits the data—the higher (less negative) the value, the better the fit. The model successfully converged in 5 iterations, meaning it found an optimal solution based on the data.

These results indicate that features like x1​ and x3 significantly impact the likelihood of the outcome, while others, such as x2​ and x7​, have minimal effect. Although the model captures some patterns in the data, the modest pseudo R-squared value suggests there is room for improvement in predicting the outcome, perhaps by including more informative features or refining the model further.

***Code for Classification Report:***

***Output:***

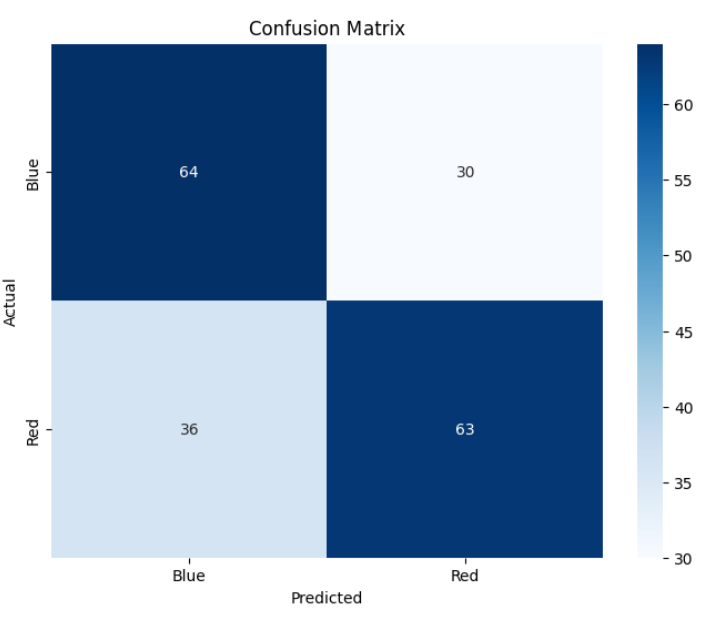
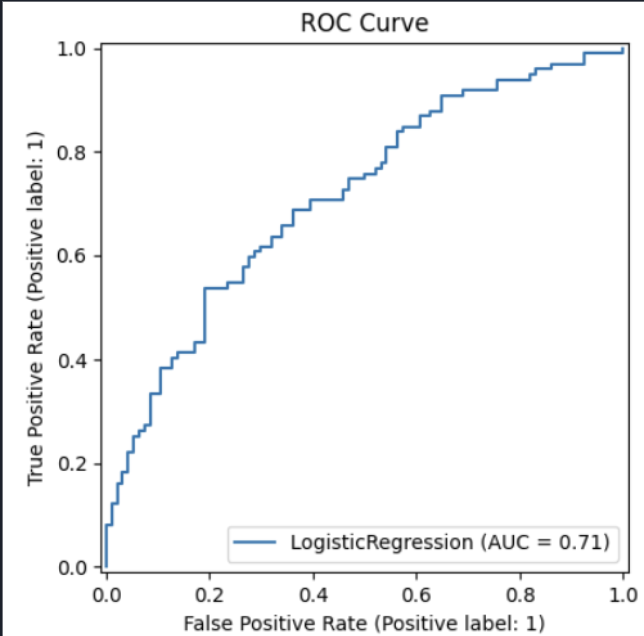


This classification report evaluates the performance of the logistic regression model on a binary classification task, where the two classes are labeled as 0 and 1. For class 0, the model achieves a precision of 0.64, meaning 64% of the instances predicted as class 0 are correct, and a recall of 0.68, indicating the model correctly identifies 68% of the actual class 0 instances. Similarly, for class 1, the model has a precision of 0.68, meaning 68% of the predicted class 1 instances are correct, and a recall of 0.64, showing that 64% of the actual class 1 instances are identified correctly. The F1-score, which balances precision and recall, is 0.66 for both classes.

Overall, the model achieves an accuracy of 0.66, meaning it correctly classifies 66% of all instances in the dataset. The macro average, which equally weights both classes when calculating precision, recall, and F1-score, is also 0.66. The weighted average, which accounts for the number of instances in each class, results in the same value of 0.66. Additionally, the ROC AUC score, which measures the model's ability to distinguish between the two classes across all decision thresholds, is 0.7135. This indicates a moderate level of separability between the two classes. Overall, the model performs consistently across both classes but shows room for improvement in predictive performance.

***Code for ROC plot:***

***Output for ROC curve and Confusion Matrix:***



The confusion matrix provides a clear visualization of the model's performance by outlining the true positives, true negatives, false positives, and false negatives. These values form the foundation for calculating the model's accuracy and other key metrics.

The ROC curve illustrates the performance of the logistic regression model in distinguishing between the two classes 0 and 1. The curve plots the true positive rate (sensitivity) on the y-axis against the false positive rate (1-specificity) on the x-axis at various threshold values. The closer the curve is to the top-left corner, the better the model is at distinguishing between the classes.

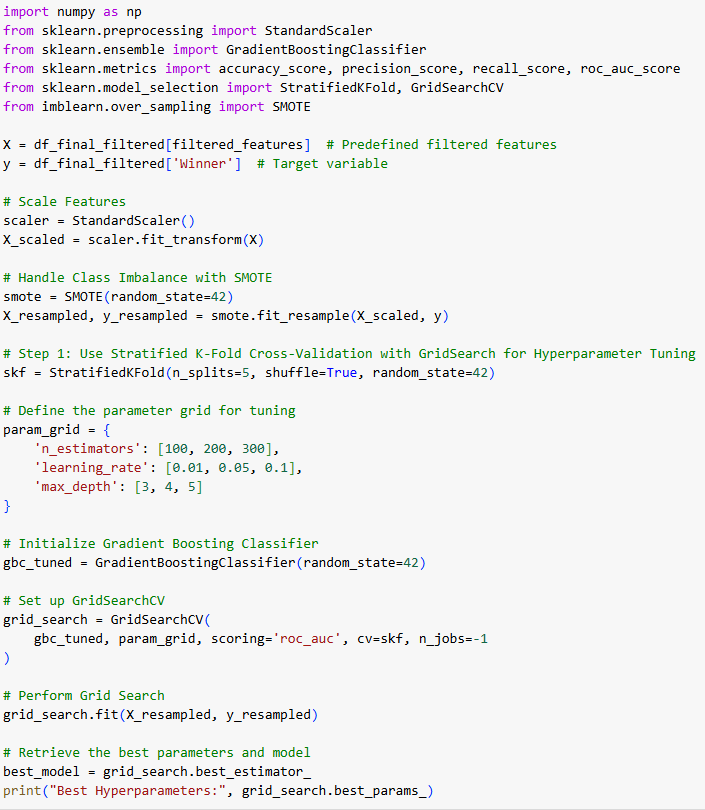
The area under the curve (AUC) is shown as 0.71, which indicates the model's ability to separate the classes. An AUC of 0.50 represents no discrimination (equivalent to random guessing), while an AUC of 1.0 represents perfect discrimination. An AUC of 0.71 suggests that the model has a moderate ability to differentiate between the two classes, performing better than random guessing but leaving room for improvement.

Overall, this ROC curve and the AUC score highlight the trade-off between true positives and false positives at different thresholds and provide a visual and numerical measure of the model's classification performance.

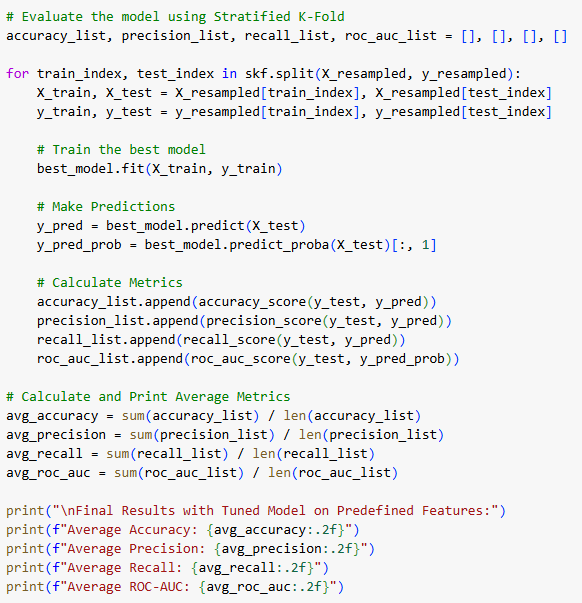
Overall, the results from the logistic regression model demonstrate moderate performance in predicting the binary outcome. The coefficients table reveals that some features, such as x1 to x3​, have a significant impact on the predictions, while others, like x2 and x7​, show little to no statistical significance. The classification report indicates that the model has a balanced performance across both classes, with a precision, recall, and F1-score of 0.66, along with an overall accuracy of 66%. While these metrics suggest the model is consistent, the relatively modest scores indicate that the predictions are not highly accurate and may benefit from further refinement. The ROC curve and the AUC score of 0.71 confirm this moderate performance, showing that the model can distinguish between the two classes better than random guessing but not with high precision. In summary, the logistic regression model provides a fair level of prediction accuracy but leaves room for improvement, either by incorporating more informative features, tuning model parameters, or exploring alternative modeling approaches.

## 

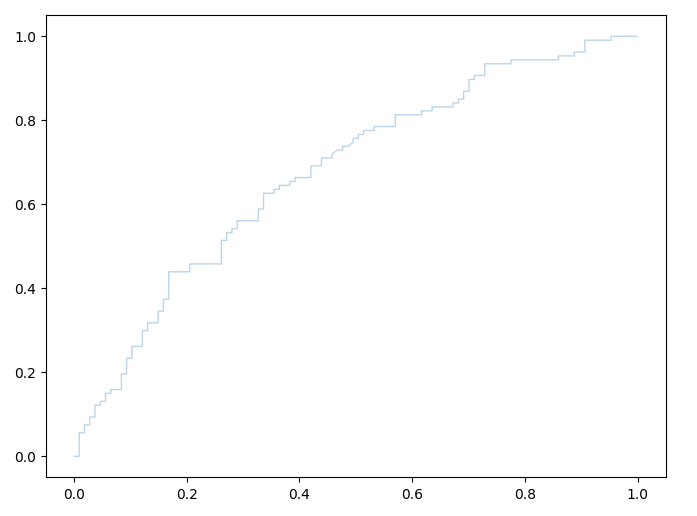
## Resampling

***Code for resampling process (explanation below):***

***Resampling code continued:***



## To improve model performance a set of instructions were made that used as many model tuning and resampling techniques as possible. To start, the features were first scaled for the gradient boosting method so that the feature scales don’t dominate the model. A SMOTE (synthetic minority oversampling technique) was also implemented. This method oversamples the minority class with synthetic samples to handle the imbalance. After handling imbalances and scaling a stratified k fold with hyperparameter tuning was implemented. The stratified k fold technique ensures that each fold in the dataset maintains the same proportion of the original dataset. For the hyperparameter tuning a grid was made with three tuning parameters; n\_estimators (number of trees in the ensemble), learning\_rate (weights factors in each tree for final prediction), and max\_depth (maximum depth of a tree). These parameters are plugged into a gridsearch setup. The grid search evaluates the combinations of hyperparameters across all k folds. It uses ROC-AUC scoring metric to determine how well the model discriminates between classes. The n\_jobs = -1 tuner allows for parallel computation. After grid search the model with the best combination of parameters is used and stored in the best model. The best model is trained on the training set and evaluated on the test set. The metrics are all averaged across every fold and displayed below.

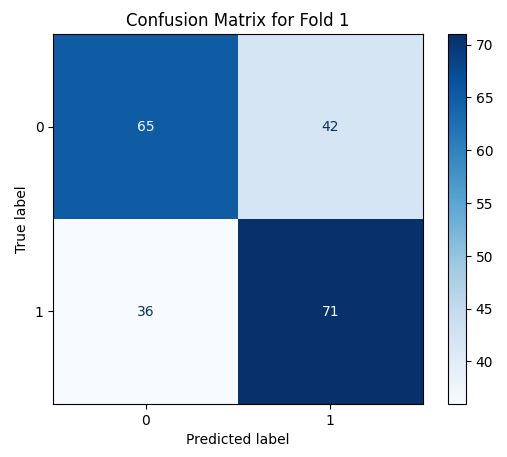


***ROC-AUC Plot:***

## 

## 

## ***Confusion Matrices for each fold:***

***Fold 1***

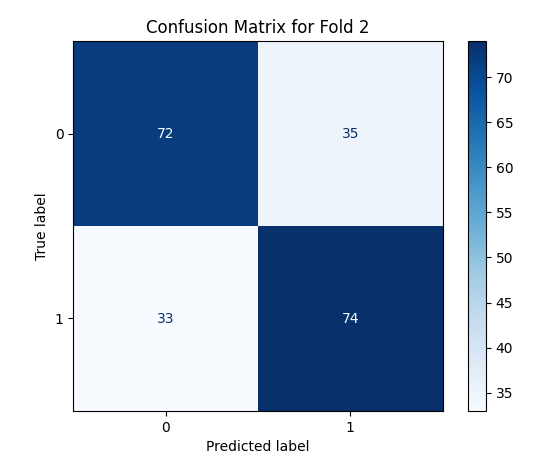
Accuracy: 64%

Precision: 61%

Recall: 64%

## 

## 

***Fold 2***

Accuracy: 68%

Precision: 67%

Recall: 69%

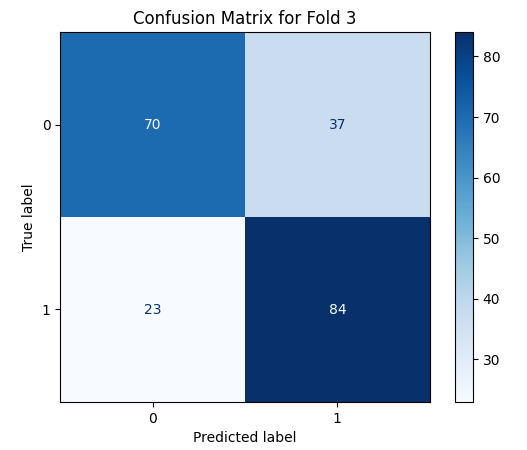
## 

## 

## 

## 

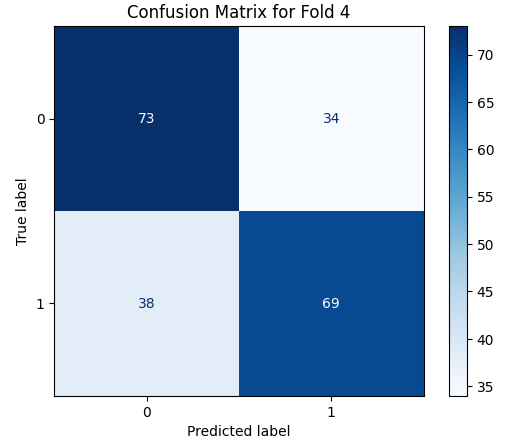
***Fold 3***

Accuracy: 72%

Precision: 65%

Recall: 75%

## 

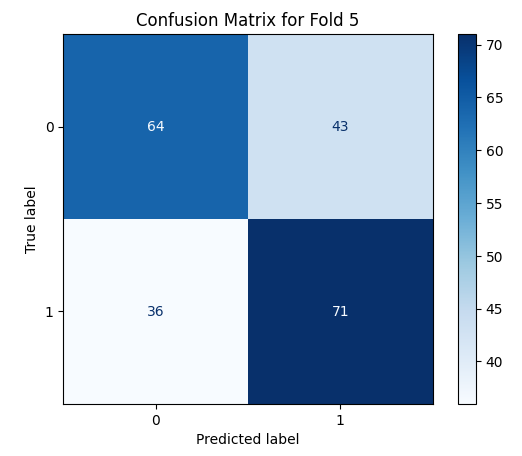
***Fold 4***

Accuracy: 66%

Precision: 68%

Recall: 66%

## 

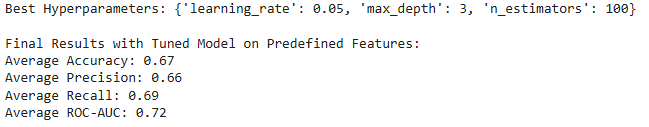
***Fold 5***

Accuracy: 63%

Precision: 60%

Recall: 64%

## 



***Averages of output:***

## 

## Conclusion

The project aimed to develop a predictive model to analyze fight statistics and determine the outcomes of UFC matches. Through a methodical process that included data cleaning, feature selection, and modeling using logistic regression, the study uncovered valuable insights and highlighted areas for improvement in predicting fight outcomes.

#### ***Main Findings:***

1. **Key Predictive Features**:
   * Features such as AvgTDDif, BlueAvgTDLanded, and RedAvgSigStrPct emerged as significant predictors. These metrics provide critical insights into a fighter’s performance and their likelihood of winning.
   * A feature selection process using Random Forest and Variance Inflation Factor (VIF) analysis effectively reduced multicollinearity and focused the dataset on the most informative predictors.
2. **Model Performance**:
   * The logistic regression model achieved an accuracy of 66%, with an ROC AUC score of 0.7135. While these results demonstrate moderate predictive ability, there is room for improvement.
   * Significant predictors like BlueAvgSigStrPct and RedAvgSigStrPct positively influenced the outcome predictions, indicating their strong contribution to the model's accuracy.
3. **Data Characteristics**:
   * Exploratory analysis revealed trends in variables such as significant strike percentages and takedown metrics, which align with observed patterns in fight outcomes.
   * However, the low correlation between some variables and the outcome (Winner) suggests that additional or alternative features might improve predictive power.
4. **Challenges**:
   * Imbalanced data and low pseudo R-squared values (0.1215) indicated a need for further refinement in feature engineering and model tuning.
   * The limited scope of available features, primarily focusing on fight statistics, may have constrained the model's ability to fully capture the complex dynamics of UFC matches.

### **Implications:**

* The findings provide actionable insights for athletes, coaches, and analysts in assessing the critical performance metrics that influence fight outcomes.
* By identifying and quantifying key predictors, the study contributes to the broader domain of sports analytics and can be a foundation for similar analyses in other combat sports.

## 

## 

## 

## Suggestions for Improvements and Future Work:

1. **Incorporate Additional Data Sources**:
   * Include psychological, physiological, or training-related data (e.g., fighter fatigue, pre-fight preparation) to capture aspects beyond in-ring performance.
   * Analyze historical head-to-head records and fighter styles (e.g., grappler vs. striker) for deeper insights.
2. **Explore Advanced Modeling Techniques**:
   * Utilize machine learning models like Gradient Boosting, Support Vector Machines, or Neural Networks, which may capture non-linear relationships better than logistic regression.
   * Implement ensemble techniques or hybrid models to improve prediction accuracy.
3. **Enhance Data Quality**:
   * Address data imbalance issues through techniques such as oversampling, undersampling, or synthetic data generation (e.g., SMOTE).
   * Collect larger and more diverse datasets, spanning multiple years and including international fights.
4. **Refine Feature Engineering**:
   * Introduce interaction terms or composite features to capture complex relationships (e.g., combining significant strikes with takedown metrics).
   * Perform temporal analysis to examine trends over time (e.g., improvement in fighter performance with experience).
5. **Evaluate Model Interpretability**:
   * Use SHAP (Shapley Additive Explanations) values to interpret and explain feature contributions to predictions, improving transparency for end-users.
6. **Broaden the Application Scope**:
   * Adapt the framework for predicting outcomes in other sports or competitive scenarios, leveraging the modularity of the analytical pipeline.

### 

### **Final Thoughts:**

This project successfully demonstrated the application of data science techniques in sports analytics, uncovering meaningful predictors and yielding a moderately accurate model. By addressing current limitations and expanding the dataset and modeling approaches, future iterations of this study have the potential to achieve higher predictive accuracy and broader applicability.

## References

* <https://www.geeksforgeeks.org/data-visualization-with-pairplot-seaborn-and-pandas/>
* <https://www.statology.org/sklearn-classification-report/>
* <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>
* <https://www.appstate.edu/~whiteheadjc/service/logit/intro.htm>
* <https://ca.novibet.com/blog/top-best-ufc-fighters-of-all-time>
* <https://www.givemesport.com/the-10-greatest-ufc-womens-fighters-of-all-time-ranked/>
* <https://talksport.com/mma/1746424/greatest-ufc-fighters-ever/>
* <https://www.kaggle.com/datasets/mdabbert/ultimate-ufc-dataset/data> 9
* <https://www.ufc.com/news/dangerous-uriah-hall-relishes-weidman-rematch-ufc-261>
* <https://fansidedmma.com/posts/red-blue-corner-fan-guide-01hvkvq95pp8>
* <https://www.ufc.com/octagon>
* <https://online.stat.psu.edu/stat462/node/180/>
* <https://pmc.ncbi.nlm.nih.gov/articles/PMC8831439/#:~:text=The%20ROC%20curve%20is%20an,or%20absence%20of%20a%20disease>.

s