Can We Predict the Outcome of a Boxing Match?

I. Introduction

Summary Of The Problem To Be Solved:

There are many boxing enthusiasts who place bets on the outcomes of fights and make a significant amount of money off this industry. They are always looking for ways to maximize their betting success rates and optimize financial gain.

In this project I am seeking to identify biometric characteristics and performance statistics that closely correlate with the fighters win percent and can be combined to build a model that accurately predicts the outcome of a future fight. I will then use these variables to build a predictive model that matches 2 fighters and predicts a winner.

The Data Used And How It Was Acquired

I used a csv file that contained the following columns for each fighter for 5144 unique fights that took place during the years 1993 and 2019. This data file was acquired from Kaggle at the following URL:

https://www.kaggle.com/rajeevw/ufcdata#data.csv

Initial Columns and Definitions:

- KD is number of knockdowns
- SIG_STR is no. of significant strikes 'landed of attempted'
- SIG_STR_pct is significant strikes percentage
- TOTAL_STR is total strikes 'landed of attempted'
- TD is no. of takedowns
- TD_pct is takedown percentages
- SUB_ATT is no. of submission attempts
- PASS is no. times the guard was passed?
- REV is the no. of Reversals landed
- HEAD is no. of significant strinks to the head 'landed of attempted'
- BODY is no. of significant strikes to the body 'landed of attempted'
- CLINCH is no. of significant strikes in the clinch 'landed of attempted'
- GROUND is no. of significant strikes on the ground 'landed of attempted'
- win_by is method of win
- last_round is last round of the fight (ex. if it was a KO in 1st, then this will be 1)
- last_round_time is when the fight ended in the last round
- Format is the format of the fight (3 rounds, 5 rounds etc.)
- Referee is the name of the Ref
- date is the date of the fight
- location is the location in which the event took place
- Fight_type is which weight class and whether it's a title bout or not
- Winner is the winner of the fight

- Stance is the stance of the fighter (orthodox, southpaw, etc.)
- Height_cms is the height in centimeter
- Reach_cms is the reach of the fighter (arm span) in centimeter
- Weight_lbs is the weight of the fighter in pounds (lbs)
- age is the age of the fighter
- title_bout Boolean value of whether it is title fight or not
- weight_class is which weight class the fight is in (Bantamweight, heavyweight, Women's flyweight, etc.)
- no_of_rounds is the number of rounds the fight was scheduled for
- current_lose_streak is the count of current concurrent losses of the fighter
- current_win_streak is the count of current concurrent wins of the fighter
- draw is the number of draws in the fighter's ufc career
- wins is the number of wins in the fighter's ufc career
- losses is the number of losses in the fighter's ufc career
- total_rounds_fought is the average of total rounds fought by the fighter
- total_time_fought(seconds) is the count of total time spent fighting in seconds
- total_title_bouts is the total number of title bouts taken part in by the fighter
- win_by_Decision_Majority is the number of wins by majority judges decision in the fighter's ufc career
- win_by_Decision_Split is the number of wins by split judges decision in the fighter's ufc career
- win_by_Decision_Unanimous is the number of wins by unanimous judges decision in the fighter's ufc career
- win_by_KO/TKO is the number of wins by knockout in the fighter's ufc career
- win_by_Submission is the number of wins by submission in the fighter's ufc career
- win_by_TKO_Doctor_Stoppage is the number of wins by doctor stoppage in the fighter's ufc career

Outline Of The Initial Plan To Solve The Problem:

1) Identify and isolate columns containing desired metrics that can be used to create the final prediction model
2) Separate data into categories reflecting performance, biometrics, and win/loss records
3) Calculate win percent and use visualizations and statistical exploration to identify metrics that demonstrate a strong correlation with win percent
4) Using Statsmodels, formulate best fit statistical models that take a value or group of values, and predicts a fighter's probability to win
5) Create an algorithm that compares 2 fighters and predicts the outcome of a future fight using the model from step 4

II. Data Wrangling Summary

Dropped Columns With Information Not Necessary For Development Of The Final Algorithm:

There were several columns in the raw data set that contained data unnecessary for achievement of the final goal of building a predictive model. There were also many columns with redundant data. These columns were removed. The result was a cleaner and more condensed data set with relevant data. Total number of columns was reduced from 145 to 18:

- Columns containing counts of unsuccessful strike attempts were dropped, but columns containing successfully landed strikes were retained
- Break-outs of strike type by individual body part were dropped, but columns containing average head strikes and average total strikes were retained
- Individual win type break-outs were dropped, but columns containing total wins and losses were retained
- All columns containing entirely NaN values or 0 values were dropped

Columns containing opponent data were dropped to isolate metrics pertaining to the performance and characteristics of one fighter in each row. I was most interested in looking at the performance and biometrics of the winner rather than the damage inflicted onto the winner by their opponent. In order to do this, the following DataFrame manipulation was performed.

- Split Data frame into 2 separate Data frames - R winners and B winners
- Dropped the loser data columns from each data set (dropped B data from R winners and dropped R data from B winners)
- Re-assigned column names to each Data frame to create consistent column names between the separated Data frames
- Concated the separated data frames vertically under the common column names
- The final result was a Data frame with one fighter per row

Final Columns:

Column Title	Object Type
Fighter	5061 non-null object
current_lose_streak	5061 non-null int64
current_win_streak	5061 non-null int64
avg_head_landed	4232 non-null float64
avg_KD	4232 non-null float64
avg_sig_str_landed	4232 non-null float64
avg_td_landed	4232 non-null float64
avg_total_str_landed	4232 non-null float64
longest_winning_streak	5061 non-null int64
losses	5061 non-null int64
total_title_bouts	5061 non-null int64
win_by_ko/tko	5061 non-null int64
wins	5061 non-null int64
stance	4946 non-null object
height_cms	5059 non-null float64
reach_cms	4768 non-null float64
weight_lbs	5059 non-null float64
age	4999 non-null float64

Dropped all null values:

Null values were all clustered in rows containing greater than 6 Null values. This signified rows containing records of unreliable data with questionable accuracy. These rows were dropped entirely in order to maintain the integrity and validity of the sample.

Results of remaining records after dropping null values:

Column Title	Object Type
Fighter	3990 non-null object
current_lose_streak	3990 non-null int64
current_win_streak	3990 non-null int64
avg_head_landed	3990 non-null float64
avg_KD	3990 non-null float64
avg_sig_str_landed	3990 non-null float64
avg_td_landed	3990 non-null float64
avg_total_str_landed	3990 non-null float64
longest_winning_streak	3990 non-null int64
losses	3990 non-null int64
total_title_bouts	3990 non-null int64
win_by_ko/tko	3990 non-null int64
wins	3990 non-null int64
stance	3990 non-null object
height_cms	3990 non-null float64
reach_cms	3990 non-null float64
weight_lbs	3990 non-null float64
age	3990 non-null float64

Dropped Rows With Extreme Outlier Values Resulting From Averages Taken Over Less Than 5 Fights:

- To find outliers, the mean, standard deviation, min, max, and percentiles were inspected. This was accompanied by histogram plots of columns to visualize extreme values
- All outlier data was found to be located in rows with averages taken across less than 5 total fights (sample sizes too small the cushion the impacts of very big or small values)
- All rows with less than 5 total fights were dropped. After this was done, all summary statistics across columns in the Data frame normalized

Final Data frame ready for analysis:

Column Name	Object Type
Fighter	1993 non-null object
current_lose_streak	1993 non-null int64
current_win_streak	1993 non-null int64
avg_head_landed	1993 non-null float64

avg_KD	1993 non-null float64
avg_sig_str_landed	1993 non-null float64
avg_td_landed	1993 non-null float64
avg_total_str_landed	1993 non-null float64
longest_winning_streak	1993 non-null int64
losses	1993 non-null int64
total_title_bouts	1993 non-null int64
win_by_ko/tko	1993 non-null int64
wins	1993 non-null int64
stance	1993 non-null object
height_cms	1993 non-null float64
reach_cms	1993 non-null float64
weight_lbs	1993 non-null float64
age	1993 non-null float64
total_fights	1993 non-null int64

III. Visualization and Exploration

Introduction To The Cleaned Data:

The cleaned data set consisted of 1,993 rows, each containing data for one boxer who won a fight during the years 1993-2019. There were 18 data columns for each row containing data related to performance statistics, win and loss records, and biometric information. The column data was as follows (all averages taken over the career of the fighter):

Column Name	Description
Fighter	The full name of the fighter
current_lose_streak	The fighter's current number of subsequent losses
current_win_streak	The fighter's current number of subsequent wins
avg_head_landed	The fighter's average number of landed strikes to the opponent's head per round
avg_KD	The fighter's average number of knockdowns per round
avg_sig_str_landed	The fighter's average significant strikes landed on the opponent per rou nd
avg_td_landed	The fighter's average takedowns of the opponent per round
avg_total_str_landed	The fighter's average total strikes landed per round
longest_winning_streak	The fighter's longest career losing streak
losses	The fighter's total career losses
total_title_bouts	The fighter's total career title bouts
win_by_ko/tko	The fighter's total career wins by knockout
wins	The fighter's total career wins
height_cms	The fighter's height in centimeters
reach_cms	The fighter's reach in centimeters
weight_lbs	The fighter's weight in lbs.
age	The fighter's current age

total_fights	The fighters total career wins + losses

Exploration and Visualization Steps:

The exploration was started with using the Python df.describe() function on the Data frame to view the summary statistics for each metric. This table shows the mean, standard deviation, min, max, and percentiles (25%, 50%, 75%).

Histogram plots of each individual metric were created to gain an initial understanding and intuition of the midpoint, deviation, and spread of the raw data. Most of the performance metrics demonstrated an right tailed distribution shape with the peak located approximately between the 25% and 50% quartiles. The biometric data showed a more normal distribution but still showed some definite preferential values and a slight right tailed shape.

I next wanted to gain insight into the relationships that existed between all the metrics and the possible correlations between each metric and the win percent. I started with creating a series of 3 pair plots. These showed some relationships among win/loss metrics, however, did not show insightful correlations between different data groups (performance metrics vs. biometric data vs. win/loss metrics).

I followed the pair plots with scatter plots of each metric vs. win percent (win percent being the value I eventually wanted my model to accurately predict). The results of these visualizations determined my path for statistical analysis.

IV. Statistical Investigation and Summary of Findings

The desired outcome of working with this data set was to find a linear model that could be used to predict the chance of a fighter winning a fight based on the fighter's historical performance metrics and biometric attributes. The results of the visualizations did not paint an optimistic view of the predictive power of a model that could be developed so I proceeded to create a model for all metrics and then use the model summary to statistics to identify the best ones.

I first calculated the correlations of each metric to the win percent for each fighter. These were the results.

Metric	Correlation with Win Percent
Current Lose Streak Correlation with Win Percent	-0.45444229961822463
Current Win Streak Correlation with Win Percent	0.5935028985617277
Head Strikes Landed correlation with Win Percent	0.2259392985933221
Knockdowns correlation with Win Percent	0.2127045354102693
Significant Strikes Landed Correlation with Win Percent	0.20255800847914524
Average Take Downs Correlation with Win Percent	0.23771622131399475
Average Total Strikes Landed Correlation with Win Percent	0.17737205092568312
Longest Winning Streak Correlation with Win Percent	0.6463959923883718
Total Tile Bouts Correlation with Win Percent	0.2612907181416824
Total Wins By Knockout Correlation with Win Percent	0.286489944539576
Height Correlation with Win Percent	0.06059615713029151

Reach Correlation with Win Percent	0.12333530695772818
Weight Correlation with Win Percent	0.05641010603308878
Age Correlation with Win Percent	-0.0882501053360032
Total Fights Correlation with Win Percent	0.0354733049856339

The correlations between performance metrics and win percent ranged from 0.18 to 0.24. The correlations between biometric attributes and win percent ranged from -0.09 to 0.12. These correlations did not indicate a strong relationship between the potential of a fighter to win a fight and their historical performance or biometric attributes.

I used Python's Statsmodels package to create a best fit liner model for each metric using the ordinary least squares method. Unfortunately, the strongest models that could be created still resulted in large residual differences between the model predictions and the data and small R-squared values. The log-likelihoods were also negative. While the models predicted "conceivably realistic" win percentages, they did not closely reflect the actual data off which they were created.

Conclusions from statistical analysis

There correlations between the metrics observed and win percentages of the fighters was not strong. This was reflected in the predictive strength of the prediction models that resulted. The strongest correlations were found between win/loss metrics and win percent; however, this could be intuitively determined by most people. The real usefulness of a predictive model would be extracted from its ability to determine the probability of a fighter winning a fight based on historical performance metrics or physical characteristics rather than win/loss history. This statistical analysis demonstrated a need for greater complexity of measurement to improve the ability to confidently predict the eventual winner of a fight with accuracy.

My recommendations would be to focus more on the effectiveness of a fighter's punch and accuracy of their landing. For example:

- Using sensor technology to measure impact forces of a fighter's punch landing
- Measurement of the fighter's accuracy in making impact at the primary body sites of greatest vulnerability
- Swing speed and avoidance reaction time

These types of measurement would better determine the effectiveness of a fighter outside of just general frequency. General frequency metrics could be considered poor predictors of a fighters potential to win a fight due to the fact that many ineffective punches will never deliver the same effective result as one strong, forceful, accurate, and impactful blow.

Effectiveness measurements such as the ones suggested above would also eliminate the need to measure general biometric data because, determination of effectiveness would always supersede biometric data in terms of importance in a predictive model. For example, and very small but effective fighter would make his size irrelevant.

V. XGBoot Modeling for Final Results

I first created a simple XGBoost model without hyperparameter tuning and after minimal data preprocessing to gain a better understanding of the accuracy of prediction that would result from the default settings on the raw data. I also needed a baseline off which to guide my future hyperparameter tuning decisions. For my model I used trees as my base learners and mean squared error regression as my loss function. I used an 80/20 train and test data split, trained the model on the training set and tested the model on the testing set. My initial number of boosting rounds was 5. This produced an RMSE of 14.4.

My next step was to perform cross validation using 4 folds, a max tree depth of 4, and 5 boosting rounds. Th results are shown in the table below. There was a slight improvement in RMSE using cross validation. The resulting RMSE was 13.8. Implementing different regularization methods (L1 and L2) and different values for alpha and lambda achieved the same resulting RMSE of 13.8.

	train-rmse-mean	train-rmse-std	test-rmse-mean	test-rmse-std
0	47.658321	0.202559	47.670566	0.879297
1	33.977088	0.140476	34.017811	0.861315
2	24.488147	0.097235	24.533840	0.845789
3	17.973477	0.065945	18.128278	0.810195
4	13.589644	0.053903	13.809209	0.725545

4 13.809209

I tried switching to implementation of L2 regularization (absolute residual) with the following results for different values of lambda

	Lambda	rmse
0	1	13.809209
1	10	14.783514
2	100	17.681025

I then tried using different values of alpha for L1 regularization (squared residual) with the following results:

	Alpha	rmse
0	1	13.829654
1	10	13.930784
2	100	14.398740

My final strategy was to use RandomizedSearchCV to identify the model parameters that would produce the lowest RMSE between the observed and predicted data. The parameter grid consisted of the following ranges

colsample_bytree: np.arange(0.3, 0.7, 0.2),
learning_rate: np.arange(0.05, 1, 0.05),
max_depth: np.arange(3, 10, 1),
n_estimators: np.arange(50, 200, 50)

I performed a cross validated randomized parameter search using 4 folds and 5 iterations (20 total fits) on the parameter grid above. The results were as follows:

Best parameters: n_estimators: 150, max_depth: 5, learning_rate: 0.3, colsample_bytree: 0.3

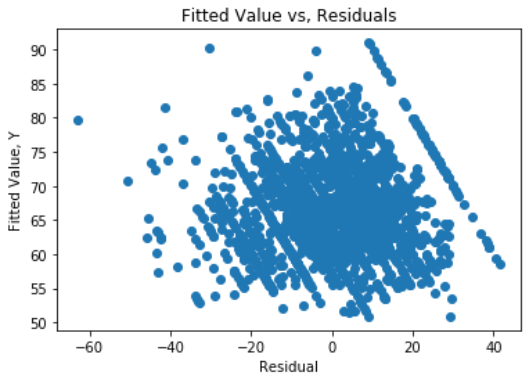
Lowest RMSE found: 6.866864381810564

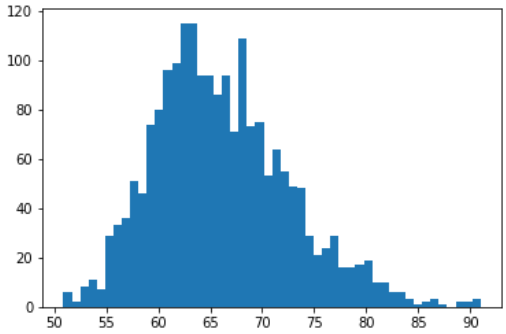
I created my final model using the following parameters

objective: reg:squarederror
colsample_bytree: 0.3
learning_rate: 0.45
max_depth: 4,
n_estimators: 50,
lambda: 1

The final RMSE on the testing data using the finalized model was 6.7.

While the XGBoost model produced a significant improvement in the RMSE, I would still use caution with using the model to make meaningful predictions. The challenges experienced during statistical analysis and attempted fitting to a linear regression model indicate large variances within the features, weak correlations between the features and the targets, and large residuals with basic linear regression predictions. To further illustrate this point I created a linear regression model using fight statistics with StatsModels. I then created plots of the distribution of the predicted values and the residuals vs. predicted values.





As can be seen in these graphs, the model predictions were reasonable without outliers, however the residuals were obscene.

My final conclusions to end this project remain the same as the conclusions I came to after completion of statistical analysis of the dataset

The correlations between the metrics observed and win percentages of the fighters were not strong. This needs to be considered in the confidence granted to the prediction models created, regardless of the strength of the techniques used. Bad data in will always create bad data out. The statistical analysis demonstrated a need for greater complexity of measurement to improve the ability to confidently predict the eventual winner of a fight with accuracy.

My recommendations would be to focus more on the effectiveness of a fighter's punch and the accuracy of their landing. Examples data collection techniques would include:

- Using sensor technology to measure impact forces of a fighter's punch landing
- Measurement of the fighter's accuracy in making impact at the primary body sites of greatest vulnerability
- Swing speed and avoidance reaction time

These types of measurement would better determine the effectiveness of a fighter rather than just general frequency. General frequency metrics could be considered poor predictors of a fighters potential to win a fight due to the fact that many ineffective punches will never deliver the same effective result as one strong, forceful, accurate, and impactful blow.

Effectiveness measurements such as the ones suggested above would also eliminate the need to measure general biometric data because, determination of effectiveness would always supersede biometric data in terms of importance in a predictive model. For example, and very small but effective fighter would make his size irrelevant.