Can We Predict the Outcome of a Boxing Match?

Industry Problem

There are many boxing enthusiasts who place bets on their predictions for the outcome of a fight. There is a significant amount of money made and lost in this industry. This has led to the pursuit of ways to objectively improve predictions, maximize betting success rates, and guarantee financial gain.

The Question:

Can the outcome of a fight be predicted using

- Historical performance metrics
- Biometric measurements
- Win/loss records

Outline Of The Initial Plan To Solve The Problem

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

The Initial Data:

- Source: https://www.kaggle.com/rajeevw/ufcdata#data.csv
- Records: Measurements taken from 5144 Fights between 1993 and 2019
- Columns: 145
- Noisy Data set with many missing values, incomplete records, redundancy

Data Cleaning and Restructuring

- Dropped columns with information not necessary for development of the final algorithm
- Restructured the data frame to achieve information for 1 fighter per row
- Dropped null values
- Dropped rows with extreme outlier values resulting from averages taken over less than 5 fights

Cleaned and Ready Data

- 1993 Records, 18 Data columns
- 6 Performance Metrics
- 4 Biometric Measurements
- 8 Win/Loss History Metrics

Performance Metrics
Average Knock Downs Per Round
Average Head Strikes Per Round
Average Take Downs Per Round
Average Total Strikes Per Round
Average Significant Striker Per Round
Total Career Title Bouts

Biometric Measurements
Height
Weight
Reach
Age

Win/Loss History
Current Lose Streak
Current Win Streak
Total CareerLosses
Total Career Wins
Career Wins by Knockout
Career Total Fights
Longest Winning Streak
Win Percent

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 round
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
win_percent	The fighters total career wins divided by their total career fights

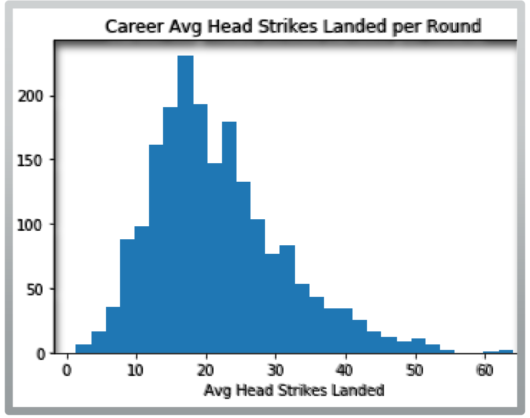
Cleaned and Ready Data

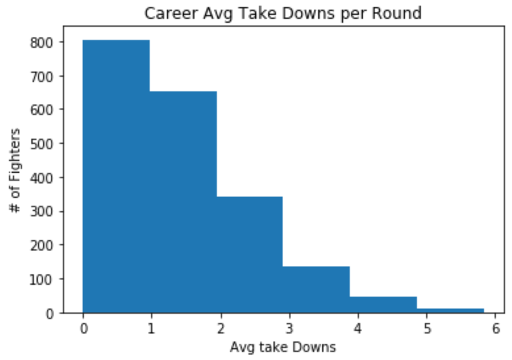
Column Name	Object Type
Fighter	1993 non-null object
current_lose_streak	1993 non-null int64
current_win_streak	1993 non-null int64
avg_head_landed	1993 non-null float64
avg_KD	1993 non-null float64
avg_sig_str_landed	1993 non-null float64
avg_td_landed	1993 non-null float64
avg_total_str_landed	1993 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
win_percent	1993 non-null float64

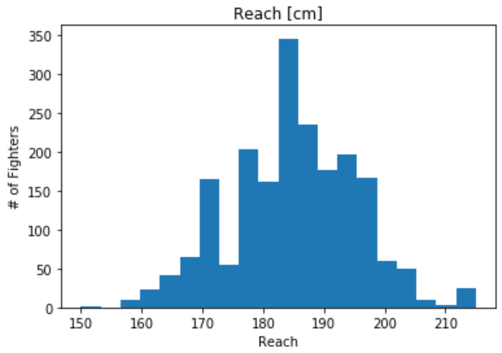
Visualization and Exploration

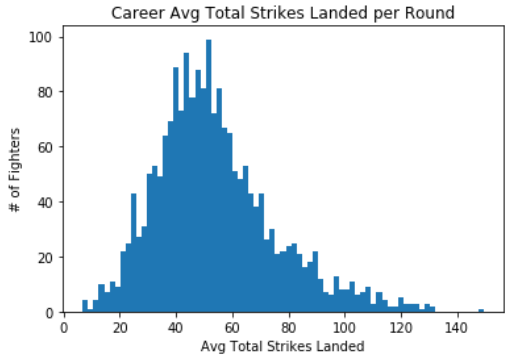
Initial Histograms:

- Insights into midpoint and spread
- Performance metrics demonstrated R tailed distribution
- Biometric measurements demonstrated centered distributions with
- Win/Loss records had weak central tendencies

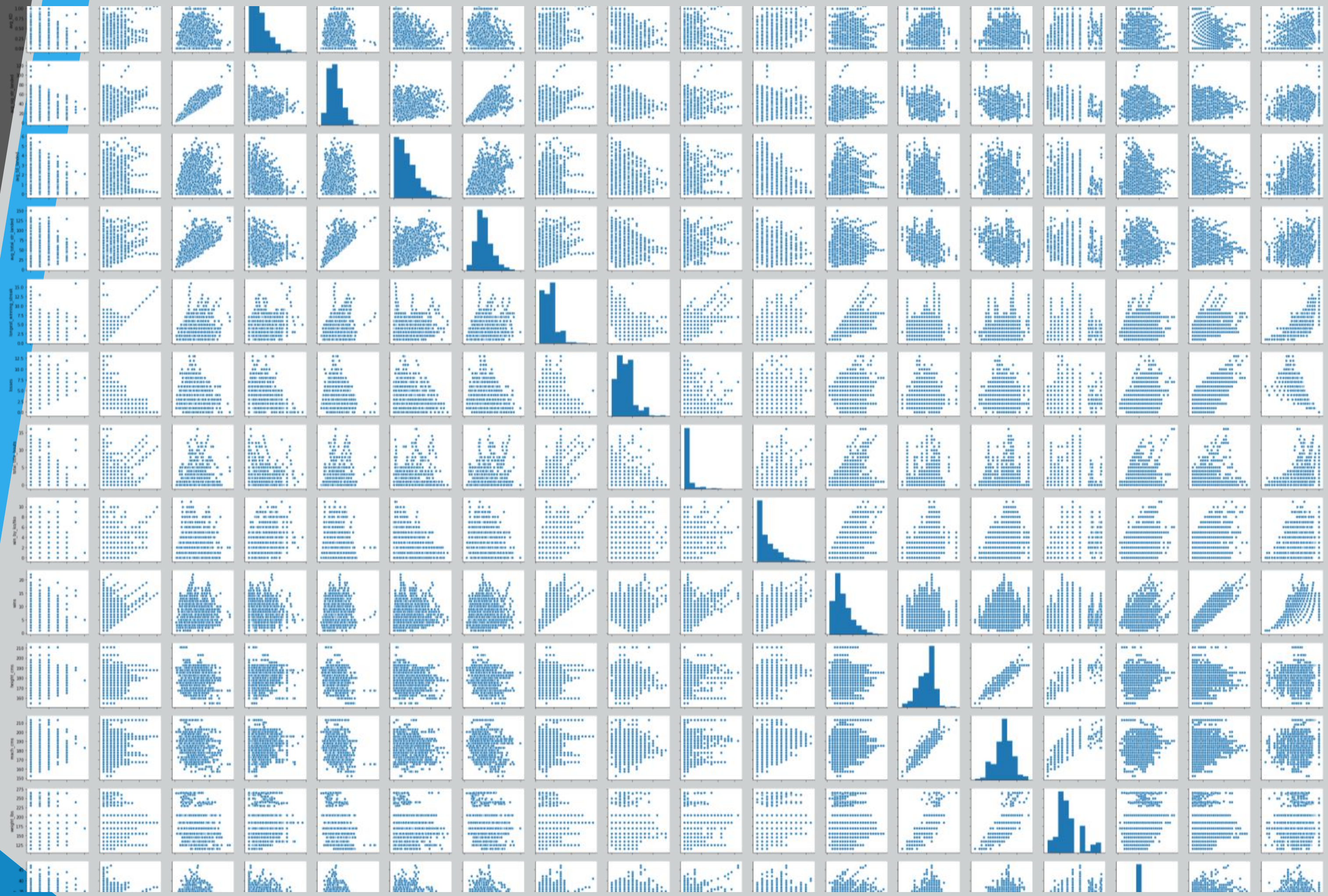








Visualization and Exploration

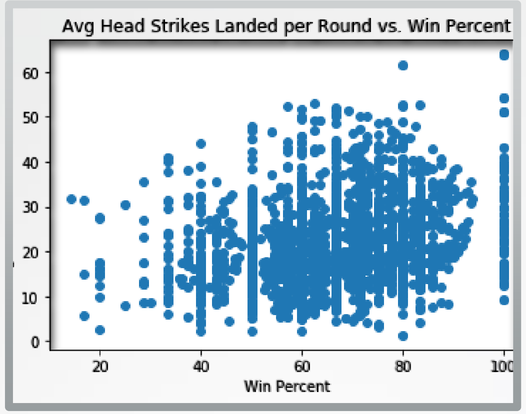


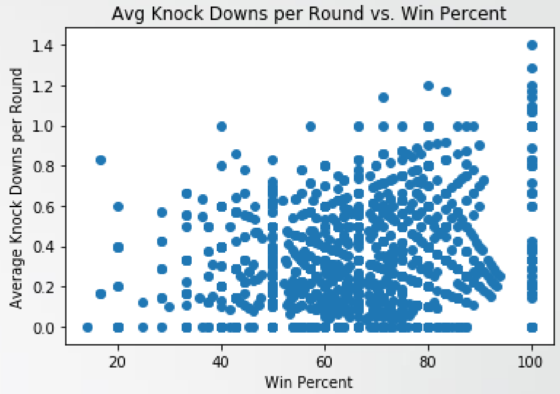
Pear Plot

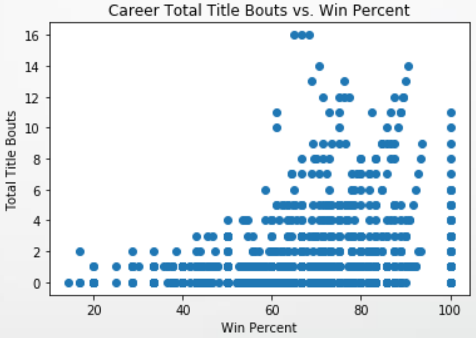
Visualization and Exploration

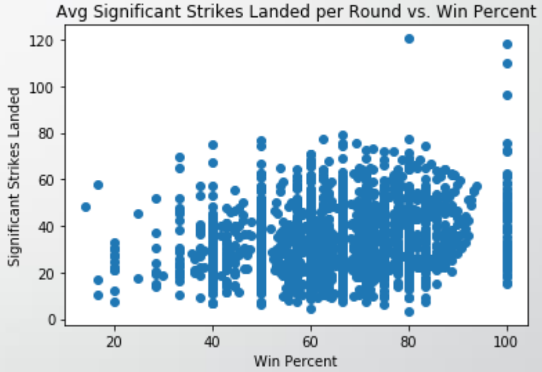
Scatter plots

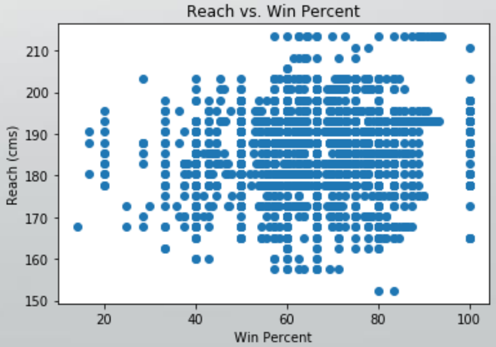
Plotted each metric vs. win percent, findings were inconclusive











Statistical Investigation: Correlations

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

Statistical Modeling

Outcomes of Ordinary Least Squares: Example Table

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            percent_win   R-squared:                       0.045
Model:                            OLS   Adj. R-squared:                  0.045
Method:                 Least Squares   F-statistic:                     94.35
Date:                Mon, 25 May 2020   Prob (F-statistic):           8.00e-22
Time:                        05:17:30   Log-Likelihood:                -8199.3
No. Observations:                1993   AIC:                         1.640e+04
Df Residuals:                    1991   BIC:                         1.641e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     62.3114      0.513    121.561      0.000      61.306      63.317
knockdowns    13.3890      1.378      9.713      0.000      10.686      16.092
==============================================================================
Omnibus:                       29.309   Durbin-Watson:                   1.923
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.733
Skew:                          -0.277   Prob(JB):                     2.12e-07
Kurtosis:                       3.252   Cond. No.                         4.51
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Initial Model Results

- 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 observed data, small R-squared values, and negative log-likelihoods
- While the models predicted "conceivably realistic" win percentages, the errors between the predicted and observed values were large.

Preliminary Conclusions

- The correlations between the individual observed metrics and win percentages of the fighters were not strong
- Consequently, confidence in the accuracy of the prediction models and usefulness of the data they produce is not strong
- Next, I will take a deeper look using different modeling techniques, but with skeptic confidence in the results

XGBoost Modeling

- The XGBoost library implements the gradient boosting decision tree algorithm
- Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction.
- The gradient descent algorithm is used to minimize the loss when adding new models.

Initial Model

- Train/test split: 80% training data, 20% testing data
- Objective: Squared Error Regression
- 5 Boosting Rounds
- Test statistic: Root Mean Squared Error (RMSE)

- Initial Resulting RMSE: 14.39

Cross Validation with XGBoost

- Objective: Squared Error Regression
- 5 Boosting Rounds
- Max Tree Depth: 4
- 4 Folds
- Test statistic: Root Mean Squared Error (RMSE)

Cross Validation Results

	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			

L2 Regularization with Parameter Search

- Used Cross Validation Model with Same Parameters as before:
  - Objective: Squared Error Regression
  - 5 Boosting Rounds
  - Max Tree Depth: 4
  - 4 Folds
  - Test statistic: Root Mean Squared Error (RMSE)

- Lambda Values: 1, 10, 100

- Results:

	Lambda	rmse
0	1	13.809209
1	10	14.783514
2	100	17.681025

L1 Regularization with Parameter Search

- Used Cross Validation Model with Same Parameters as before:
  - Objective: Squared Error Regression
  - 5 Boosting Rounds
  - Max Tree Depth: 4
  - 4 Folds
  - Test statistic: Root Mean Squared Error (RMSE)

- Alpha Values: 1, 10, 100

- Results

	Alpha	rmse
0	1	13.829654
1	10	13.930784
2	100	14.398740

Hyperparameter Tuning Using Randomized Search with 4 Fold Cross Validation

- Static Parameters
  - Objective: Squared error regression
  - 5 Boosting Rounds, 4 folds
  - Max Tree Depth: 4

- Parameters Searched
  - '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)

Hyperparameter Tuning Results and Final Model

Best parameters found:

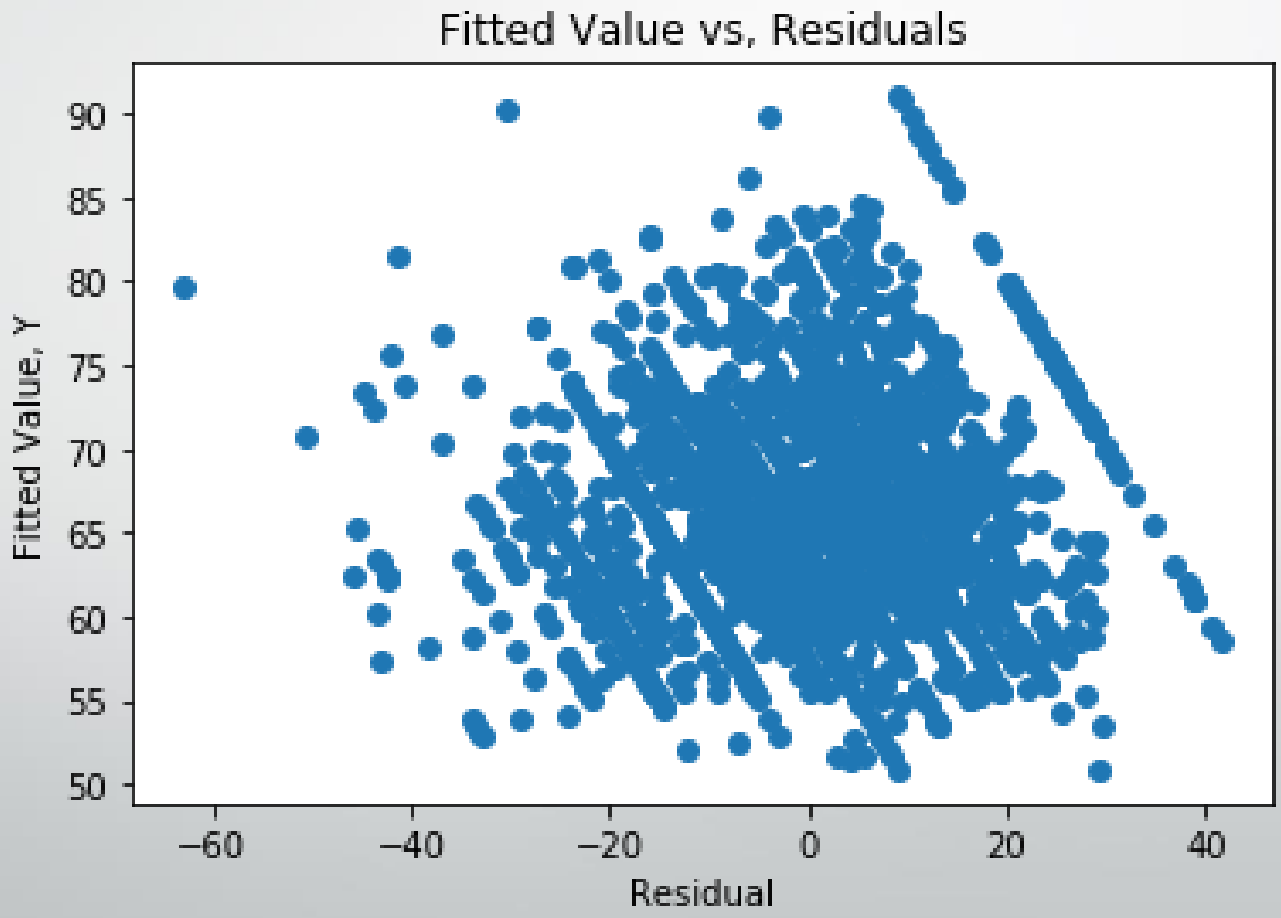
- Number estimators: 50
- Max tree depth: 4
- Learning rate: 0.45
- Sample size by tree: 0.3
- L2 regularization with lambda=1
- Number of boosting rounds: 50

Lowest RMSE: 6.65

One Final Attempt:
OLS Regression using Performance Metrics

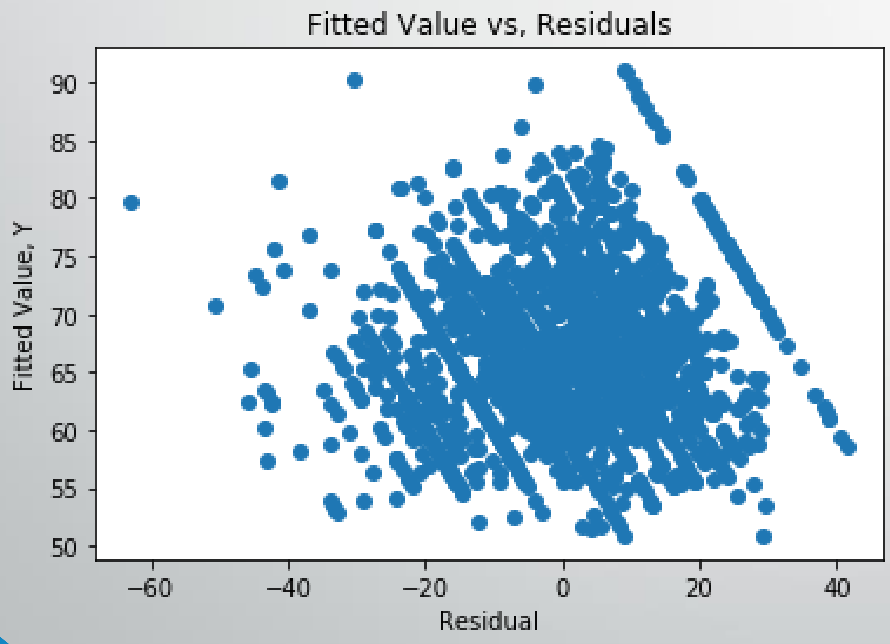
```
                            OLS Regression Results
==============================================================================
Dep. Variable:            win_percent   R-squared:                       0.184
Model:                            OLS   Adj. R-squared:                  0.182
Method:                 Least Squares   F-statistic:                     89.63
Date:                Sun, 28 Jun 2020   Prob (F-statistic):           3.33e-85
Time:                        05:00:06   Log-Likelihood:                -8042.8
No. Observations:                1993   AIC:                         1.610e+04
Df Residuals:                    1987   BIC:                         1.613e+04
Df Model:                           5
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Intercept               48.6032      1.023     47.487      0.000      46.596      50.610
avg_head_landed          0.1949      0.079      2.473      0.013       0.040       0.350
avg_KD                  18.7745      1.381     13.599      0.000      16.067      21.482
avg_sig_str_landed       0.1255      0.060      2.093      0.036       0.008       0.243
avg_td_landed            5.0141      0.359     13.958      0.000       4.310       5.719
avg_total_str_landed    -0.0601      0.025     -2.370      0.018      -0.110      -0.010
==============================================================================
Omnibus:                       37.750   Durbin-Watson:                   1.948
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               42.636
Skew:                          -0.288   Prob(JB):                     5.52e-10
Kurtosis:                       3.426   Cond. No.                         354.
==============================================================================
```

Many Values with Large Residual/Value Ratios

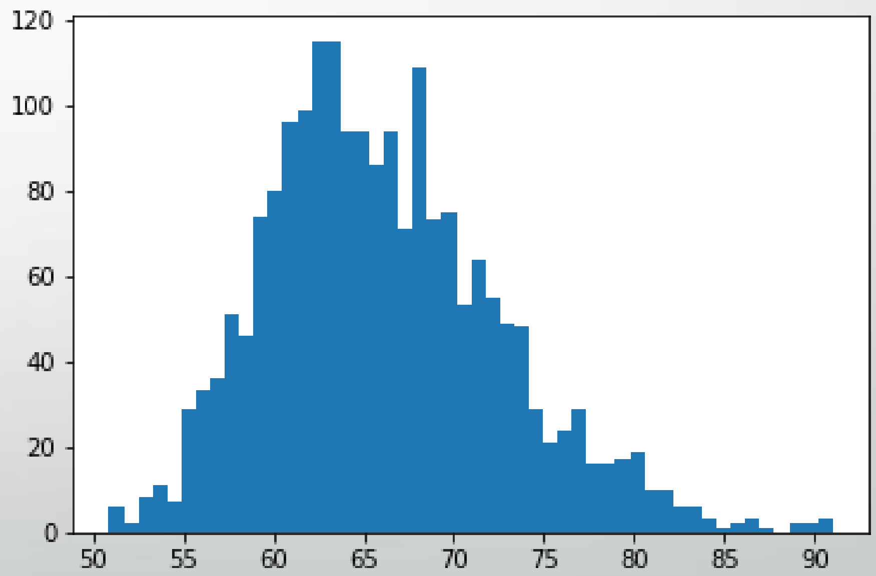


Inaccuracy of Predictions

Large Residual Values



Realistic Predicted Values



Final Conclusions

The correlations between the observed features 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.

Future Recommendations

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

Future Recommendations

- Effectiveness measurements would better determine the win potential of a fighter rather than just general strike frequency or biometric attributes
- General frequency metrics could be considered poor predictors of potential to win a fight because that many ineffective punches will never deliver the same effective result as one strong, forceful, accurate, and impactful blow
- Effectiveness measurements would also eliminate the need to consider 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.