Capstone 2: UFC Fight Analysis

What does it take to win a fight?

To find out, we will analyze data collected on these factors:

- Physical traits (e.g. age, height, and weight)

- Behaviors (e.g. take downs, submissions, footwork, clinches, and strikes)

Dataset Description

- The data set has 5144 rows/151 columns with data on each fighter in a ‘red vs blue’ format
- Each row describes a record of a fight in the UFC
- Each column describes variables including:
  - Physical traits of each fighter
  - Behaviors of fighters in the fight
  - The outcome of the fight (red win/blue win/draw)

Data Cleaning

- To prepare the data for analysis, the following steps were taken:
  - Fights with missing data and draws as outcomes were removed reducing the number of rows to 3648
  - Differences between the quantified behavior of each fighter were taken to be used for analysis
  - The outcome of the fight was one-hot-encoded to describe whether the red fighter wins or loses
  - Applied SMOTE to balance data so red wins equal red losses

The Resulting Dataframe

The resulting dataframe contains the following variables:

- **diff_age**: the difference in age between the 2 fighters
- **diff_height**: the difference in height between the 2 fighters
- **diff_reach**: the difference in reach between the 2 fighters
- **diff_avg_body_att**: the difference in the average number of body strikes attempted per round between the 2 fighters
- **diff_avg_body_landed**: the difference in the average number of body strikes landed per round between the 2 fighters

The Resulting Dataframe

- diff_avg_clinch_att: the difference in the average number of clinches attempted per round between the 2 fighters
- diff_avg_clinch_landed: the difference in the average number of clinches landed per round between the 2 fighters
- diff_avg_distance_att: the difference in the average number of attempts to close distance per round between the 2 fighters
- diff_avg_distance_att: the difference in the average number of successful attempts to close distance per round between the 2 fighters
- diff_avg_ground_att: the difference in the average number of ground strikes attempted per round between the 2 fighters

The Resulting Dataframe

- diff_avg_ground_landed: the difference in the average number of ground strikes landed per round between the 2 fighters
- diff_avg_head_att: the difference in the average number of head strikes attempted per round between the 2 fighters
- diff_avg_head_landed: the difference in the average number of head strikes landed per round between the 2 fighters
- diff_avg_kd: the difference in the average number of knockdowns per round between the 2 fighters
- diff_avg_leg_att: the difference in the average number of leg strikes attempted per round between the 2 fighters

The Resulting Dataframe

- diff_avg_leg_landed: the difference in the average number of leg strikes landed per round between the 2 fighters
- diff_avg_pass: the difference in the average number of guard passes per round between the 2 fighters
- diff_avg_rev: the difference in the average number of reversals per round between the 2 fighters
- diff_avg_sig_str_att: the difference in the average number of significant strikes attempted per round between the 2 fighters
- diff_avg_sig_str_landed: the difference in the average number of significant strikes landed per round between the 2 fighters

The Resulting Dataframe

- diff_avg_sig_str_pct: the difference in the average number of significant strikes protected against per round between the 2 fighters
- diff_avg_sub_att: the difference in the average number of submissions attempted per round between the 2 fighters
- diff_avg_td_att: the difference in the average number of takedowns attempted per round between the 2 fighters
- diff_avg_td_landed: the difference in the average number of takedowns landed per round between the 2 fighters
- diff_avg_td_pct: the difference in the average number of takedowns protected against per round between the 2 fighters

The Resulting Dataframe

- diff_avg_total_str_att: the difference in the average number of total strikes attempted per round between the 2 fighters
- diff_avg_total_str_landed: the difference in the average number of total strikes landed per round between the 2 fighters

Exploring the Data

Four models were tested during exploration, all of which were tuned with grid search:

- **KNN**
- **Logistic Regression**
- **Decision Tree**
- **Random Forest**

Exploring the Data

KNN Results

Best Score: 0.7039

	Precision	Recall	F1-Score	Support
Red Wins	1.0	1.0	1.0	415
Red Loss	1.0	1.0	1.0	395

KNN Cf Matrix	Predicted Red Win	Predicted Red Loss
Actual Red Win	415	0
Actual Red Loss	0	395

Exploring the Data

Logistic Regression Results

Best Score: 0.5991

	Precision	Recall	F1-Score	Support
Red Wins	.63	.64	.63	415
Red Loss	.62	.61	.61	395

KNN Cf Matrix	Predicted Red Win	Predicted Red Loss
Actual Red Win	265	150
Actual Red Loss	155	240

Exploring the Data

Decision Tree Results

Best Score: 0.6130

	Precision	Recall	F1-Score	Support
Red Wins	.75	.76	.76	415
Red Loss	.74	.74	.74	395

KNN Cf Matrix	Predicted Red Win	Predicted Red Loss
Actual Red Win	315	100
Actual Red Loss	104	291

Exploring the Data

Random Forest Results

Best Score: 0.6698

	Precision	Recall	F1-Score	Support
Red Wins	.95	1.0	.98	415
Red Loss	1.0	.95	.97	395

KNN Cf Matrix	Predicted Red Win	Predicted Red Loss
Actual Red Win	415	0
Actual Red Loss	21	374

Which models should we look at?

The Bad

- KNN: performed the best, but the validity of the results might be affected by the SMOTE technique used to balance the data, so kNN won’t be analyzed further
- Logistic regression: performed bad and won’t be analyzed further

Which models should we look at?

The Good

- Decision tree: performed decently and will be analyzed further
  - gridsearch tuning suggested an entropy based decision tree with a max depth of 6 and a minimum leaf size of 17
- Random forest: had the best performance and will be analyzed further
  - gridsearch tuning suggested a random forest with 100 non-bootstrapped sampled entropy based trees of max depth 6, minimum leaf size of 3, and splits with minimum samples of 3

Model Selection

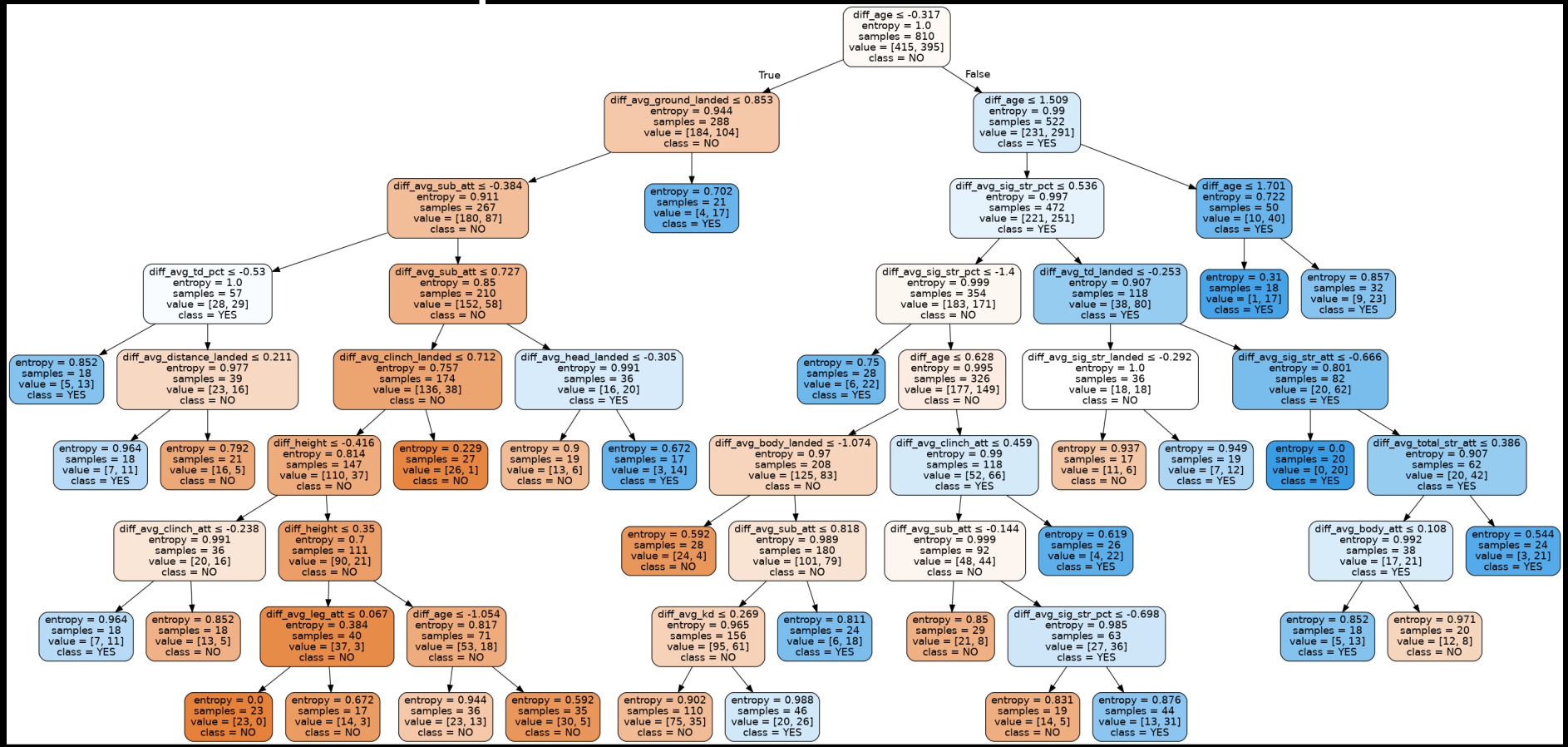
How reliable is a Decision Tree model?

- When the model predicts red wins, it's right 75% of the time
- When the model predicts red loses, it's right 74% of the time
- When red actually wins, the model is right 76% of the time
- When red actually loses, the model is right 74% of the time

- Overall, the model is right 75% of the time

Model Selection

Decision tree plot



Model Selection

Decision tree notable outcomes

- There seem to be few pathways that lead to relatively low entropy indicating situations where the model is accurate including examples such as:
  - Red fighter is younger → has not too many more ground strikes landed → has less submissions attempted →has not many less clinches landed → is shorter → red loses
  - Red is the same age as blue → has protected against less significant strikes →has attempted more takedowns → has attempted more significant strikes → red wins

Model Selection

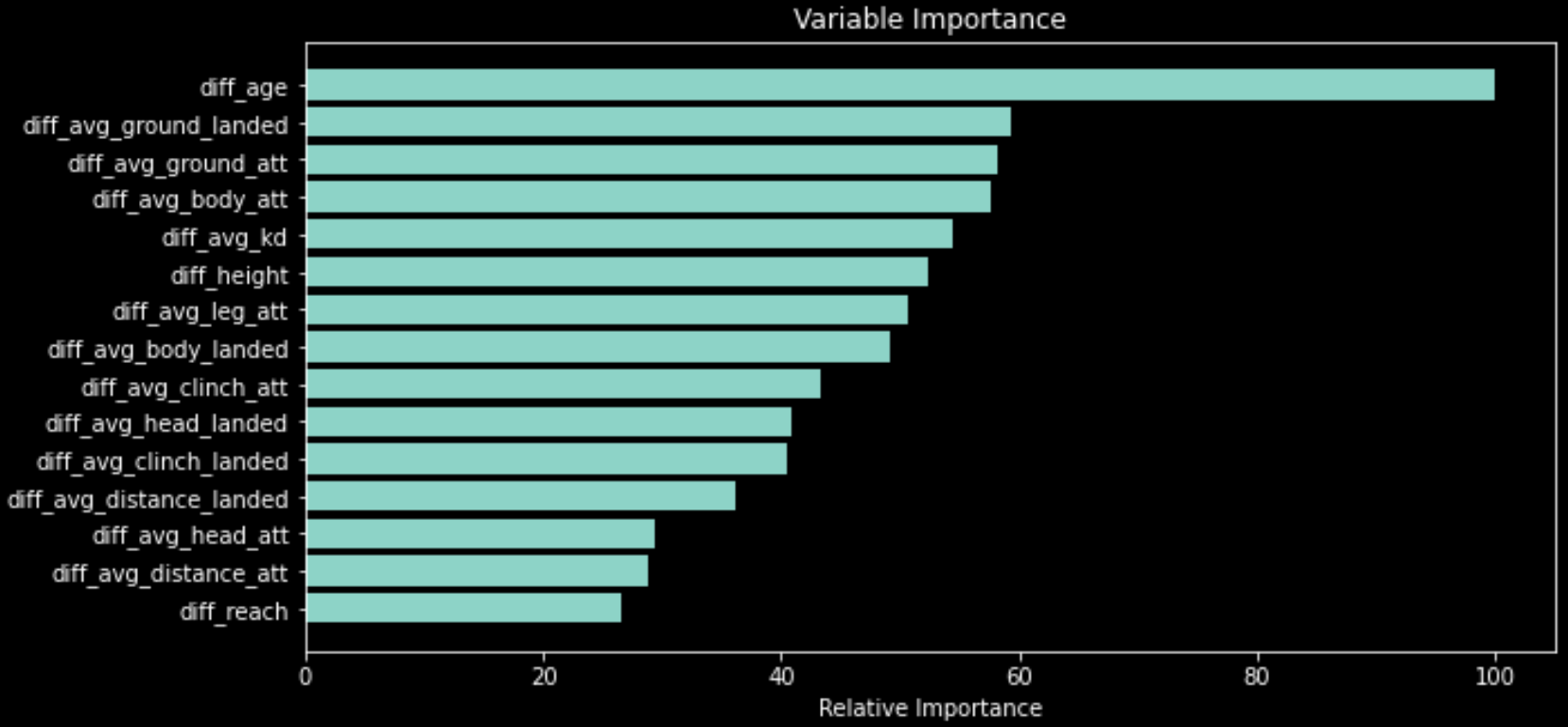
How reliable is a Random Forest model?

- When the model predicts red wins, it’s right 95% of the time
- When the model predicts red loses, it’s right 100% of the time
- When red actually wins, the model is right 100% of the time
- When red actually loses, the model is right 95% of the time

- Overall, the model is right 97% of the time

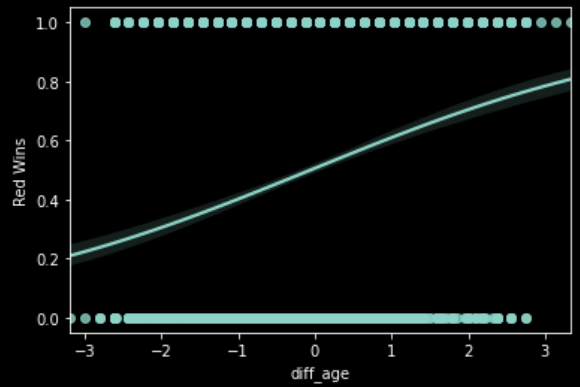
Model Selection

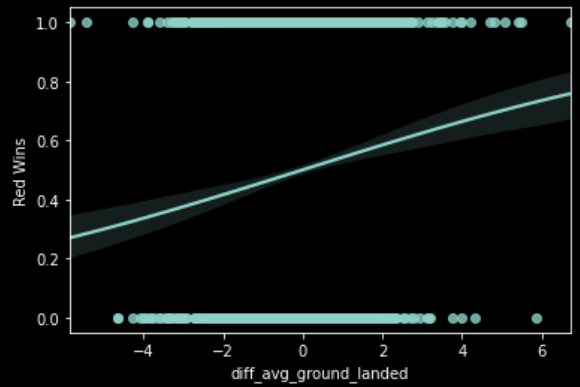
Variable importance plot

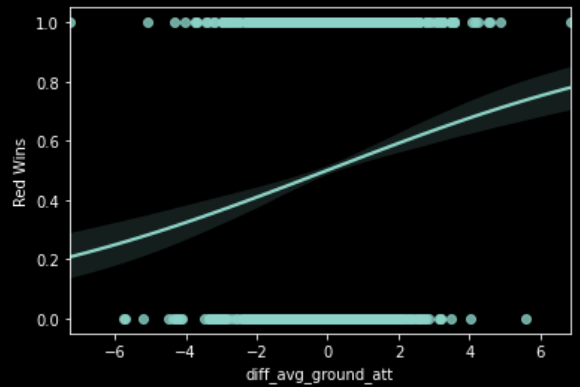


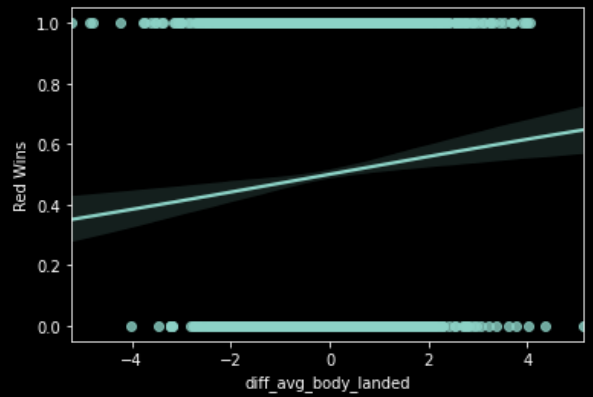
Model Selection

Log regression plots of top 4 important variables









What have we learned?

The results of our analyses show that:

- Fighters that are older than their opponents tend to be more likely to win suggesting that experience tends to triumph over youth
- Many fights are won on the ground as fighters who strike grounded opponents more often tend to have a greater chance of winning

What can we do better?

For future research we can try:

- Different techniques for balancing the data
- Feature selection algorithms for dimension reduction
- More classification models such as boosting, naive Bayes', SVM, ect.

Remember to train hard and never stop testing yourselves!

Thanks for attending!