UFC Correlations

Jordan Meidinger Jayden Rosenau

Using the database provided by Kaggle which includes fights from 1993 to 2019 with 5,144 matches and 44+ columns such as height, reach distance, number of strikes, take downs, etc. The main objective is finding the correlation between different attributes that contribute to the win of the fight. We will find the main attributes by testing smaller subsets that present a higher precision for the model.

Data Preparation

We started off by taking the raw data and splitting it up into better items that we can work with such as dropping referee, draws, names of fighters, location, and date. These columns we found insignificant in the correlation of determining the winner. Next, we started looking through the data to find the Null or empty data points to see if we could find a good replacement or remove the fight as a whole. The first feature that we found with a lot of missing data points was reach in centimeters. The data points that we do have for height and reach which we plotted to see if there was a correlation (*Figure 1*). The figure below shows that there is a positive correlation so we will replace each missing reach with the median height.

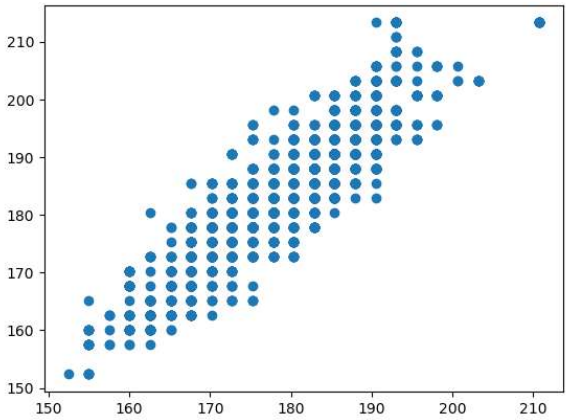

Figure 1

Another feature with a high amount of nulls was the fighter's attack stats (clinches, significant punches, kicks, and Etc…) due to not having prior information on new fighter entering the UFC. The weight classes and stance of the fighters needed to be changed from an object value to an integer. We achieved this through pandas use of concat and dummies. Next, selected types of int and float in our dataset where we split the data to training and testing. This data will be used for every model.

Random Forest (Bagging)

The first model we used was random forest which we got from SKLearn. The first test was size of 5% and using 100 estimators. The Out of the Bag score for the parameters was a 67.3% with a prediction accuracy of 63.8%. This tests most important features included opponent head attacks landed, average punch distance, age, significant strike landed and attempted. In a attempts to make improvements to the accuracy and prediction score we changed the estimators to 1000 and then 2000. By changing the estimators, we had little to no change in the accuracy or Out of the Bag score. With the distribution of blue(1144) and red(2268) winner we decided to do over sampling. Where the Out of the Bag score was 85% and a accuracy of 64%.

Without Sampling

Sample size	5%	5%	10%	10%	15%	15%
Num of estimator	100	1000	100	1000	100	1000
Out of the Bag	67.3%	68.1%	66.8%	68.5%	67.1%	68.1%
Accuracy	63.8%	62.7%	66.1%	67.2%	66.4%	67.3%

With Sampling

Sample size	5%	5%	10%	10%	15%	15%
Num of estimator	100	1000	100	1000	100	1000
Out of the Bag	84.5%	86.4%	84.0%	87.0%	83.9%	86.3%
Accuracy	65.5%	62.2%	63.0%	66.6%	66.6%	66.7%

In conclusion of Random forest algorithm accuracy is around 62% to 67% without the bag better with sampling but does not increase accuracy much.

AdaBoost (Boosting)

The second model we used to test our database of fights was AdaBoost provided by SKLearn. The reason we picked this algorithm is because we have a large amount of weak data points which AdaBoost will take and make it into one strong learner. Due to the fact that we have some many inputs the AdaBoost has a hard time determining the good features. The features that were recognized as important were signification strikes landed, age, head attacks landed for the red fighter. After testing different sizes and number of estimators it's about the same as the random forest model (63%-69% accuracy) but, had an increase of 2%. On the next page you can see the result of AdaBoost (num of estimators = 100)

Sample size	5%	5%	10%	10%	15%	15%
Learning rate	50%	100%	50%	100%	50%	100%
Accuracy	63.3%	64.4%	68.9%	66.1%	64.9%	65.1%

In conclusion of AdaBoost algorithm accuracy is around 63% to 69%, which is the best score so far.

Naïve Bayes (Gaussian)

The third model we used to test our database of the fights was Naïve Bayes provided again by SKLearn. This is a very popular algorithm used in the data mining eco system. The algorithm is a probabilistic classifier which is a classifier that is able to predict, given an observation of an input. We included this algorithm because, the dataset is small but, complex.

Sample size	5%	10%	15%
Accuracy	61.1%	59.4%	60.3%

The reason that the Naïve Bayes does not produce well is the fact that the majority of the data is numerical and is not multi-class.

Results

In conclusion if a user had no prior knowledge of fighting in the UFC, they could have a greater knowledge by using stats to better predict the outcome of any future fights. This although is not perfect system for any betting man or women, due to the fact that the probability is only slightly greater than flipping a coin. The limitations of our project is that we don't have full records of every fight which gives us null data points for fighters that have never fought previously in the UFC. The reach of the fighter is an important feature that we had to improvise that could affect the results. The stats of the fighters were also the average stats that lead up to the fight. This causes to have dynamic change for each fighter as could skew in one feature compared to another. Example Johns Jones could have a fight with a large amount kicks in 5 rounds with him winning by decision, so his kick average would increase immensely. The next fight could be a submission in one round with no kicks greatly decreasing the kick average. With these both being wins you can see how the data itself can be miss leading.

	Sample Size	Learning rate	Accuracy
Random Forest without sampling	15%	None	67.3%
Random Forest with sampling	15%	None	66.7%
AdaBoost	10%	50%	68.9%
Naïve Bayes	5%	None	61.1%