Predicting UFC fight outcomes using Random Forest

Harsha Kalidindi
University of Central Florida
MAP 4112 Fall 2020

1 Introduction

Sports analytics is a field that has seen tremendous growth in the past decade. Although organizations in major sports such as Basketball, Football, and Hockey use analytics extensively, the same hasn't been as true for fighting organizations such as the UFC. Additionally, sports betting is a $155 billion dollar industry, with combat sports ranking at the top of the industry.

In this project we will briefly outline a random forest model and then use a random forest classifier to predict UFC fight outcomes retroactively.

Dataset

The dataset was scraped from ufcstats.com, a website that collects official data from fights dating back to 1933. Original dataset contained information of 5144 fights dating back to 1993. Samples with missing information were removed and the remaining data was cleaned of irrelevant data such as time, location, referee, etc., and organized into strictly numerical categories.

After cleaning, we are left with a 5061x31 matrix. We have 5061 complete samples each with 31 features. Since we are predicting the winner of the fight, we have 30 features/independent variables that we will use to predict the winner.

In implementation of the model a 75%:25% training:testing split was used.

Below is an example of what a sample looked like after cleaning.

(Full list of features available on github)

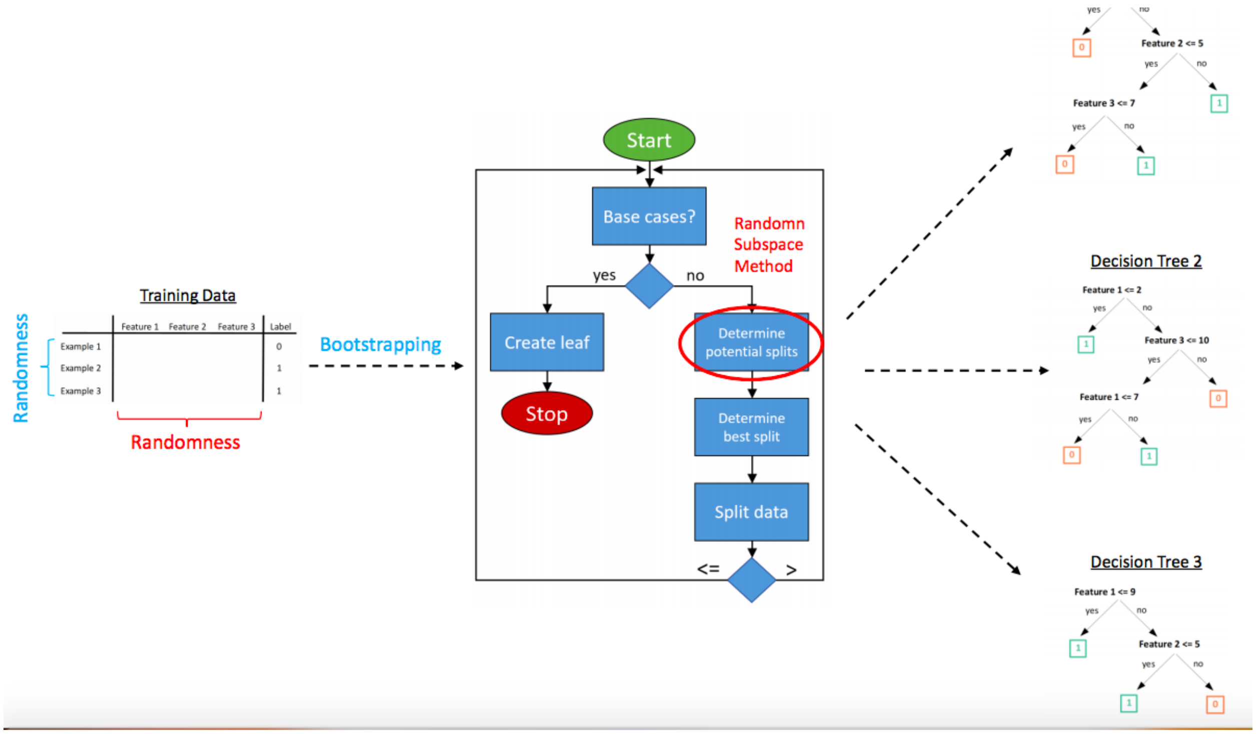
R_KD	B_KD	R_SIG_STR.	B_SIG_STR.	...	B_CLINCH	R_GROUND	B_GROUND	Winner
0	0	90	57	...	2	26	1	0

y

2 Notation and assumptions

Random Forest Background

Visual of how a Random Forest is implemented.



Essentially we are introducing randomness to a collection of decision trees. We create bootstrapped data sets by randomly sampling from the original data and then use the random subspace method and Gini impurity / Gini Gain to choose which feature to use as a node on a particular decision tree.

Gini impurity measures how ‘impure’ a feature is, specifically the probability of incorrectly classifying an element in the dataset.

Gini impurity	Classification	$\sum_{i=1}^{C} f_i(1-f_i)$	f_i is the frequency of label *i* at a node and *C* is the number of unique labels.

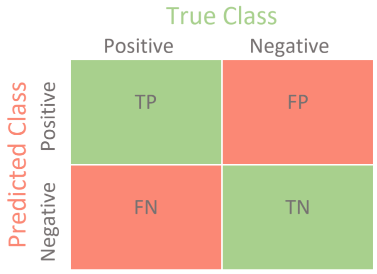
Gini Gain: calculated by subtracting the weighted impurities of the branches from the original Gini impurity. This is maximized to choose which feature to use at a node

Accuracy

Standard way of measuring accuracy: y and ŷ are predicted and true values for each position, respectively

$$accuracy\left(y, \hat{y}\right) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1\left(\hat{y} = y_i\right).$$

Another way of calculating accuracy is through a Receiver Operating Characteristic (ROC) curve. This is calculated by using True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). We create a confusion matrix and graph the curve using 1-'Specificity' along the x-axis and 'Sensitivity' along the Y axis

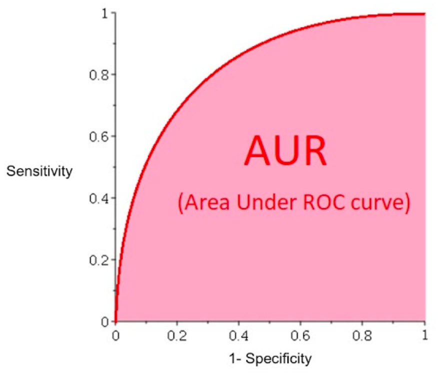


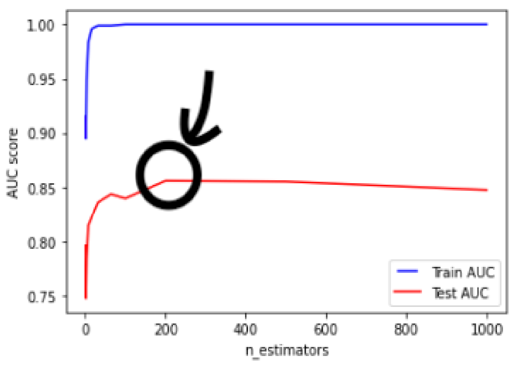
		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

We then take the area under this curve (AUC) as another measure of accuracy. This method was used to determine ~200 decision trees gave our Random Forest the best accuracy.





Feature importance

Gini importance (not to be confused with Gini Impurity) is used to calculate a particular node's importance for each decision tree.

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

ni_j = importance of node j, w_j = weighted # samples reaching node j, C_j = impurity of node j, left(j) = child node from left split on node j, right(j) = child node from right split on node j

Feature importance is then calculated for each feature on a decision tree

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

fi_i = importance of feature i, ni_j = importance of node j

Feature importance is then normalized:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

And averaged over all trees:

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

T = total # of trees, $normfi_{ij}$ = normalized feature importance for i in tree j, $RFfi_i$ = importance of feature i over all trees

3 Key results

As outlined above, AUC of ROC curve was used to test Random Forests with up to 1000 decision trees. The model performed best with 200 trees, seeing a slight decline in accuracy with additional trees. Using the standard method of measuring accuracy outlined first, 89% accuracy on test data was achieved.

Feature Importance was calculated as above. Most important features and least important features listed here:

Most important features

- Ground strikes
- Knockdowns
- Significant strikes
- Head strikes

B_GROUND	7.86%
B_KD	7.41%
R_GROUND	6.01%
B_SIG_STR.	5.82%
B_HEAD	5.66%
R_HEAD	5.37%

Least important features

- Reversals
- Takedowns
- Leg strikes

B_REV	0.39%

R_REV	0.42%
R_TD	1.44%
B_TD	1.57%
B_CLINCH	1.69%
R_LEG	1.70%
B_LEG	1.82%

4 Conclusion

In this project we outlined a Random Forest model and implemented a Random Forest Classifier to predict fight outcomes retroactively. The model was able to predict fight outcomes with 89% accuracy using 200 trees. There aren't many references of fight prediction models to compare to, so it's hard to judge how good this model really is. That being said, there are lots of interesting things we can do to improve this model. We can discard features with low importance, test different numbers of random features selected at each node, and use regression rather than classification. We can also use Random Forest models in parallel.

This model leads to further questions as well. There's lots of additional data on specific fighters as well as how judges score UFC bouts. . We can build models to predict fight outcomes based on statistics of individual fighters, which could have an impact on the sports betting industry. Additionally, one of the most controversial areas of the sport is judging. With the available data on how judges score bouts round by round, we can build models that keep judges accountable or impact how the sport is scored.

References

Code and complete list of features available on github:

https://github.com/harshak13/MAP4112

Python standard library:

https://docs.python.org/3/library/index.html

Images:

https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

https://www.sebastian-mantey.com/theory-blog/random-forest-algorithm-explained