Section 1: Introduction

Mixed Martial Arts (MMA) is the fastest growing competitive sport in the world. Because the fighters engage in distinct martial art disciplines (boxing, wrestling, jiu jitsu, etc.) on their feet, grappling, and on the ground, it would be interesting to implement learning algorithms on these fights to find any potential patterns as very few have tried. As input, we are using 8 key statistics collected over a fighter's career (the fighter's profile). As output we're predicting the outcome of a fight between the two fighters, as well as clustering the fighters stylistically.

Section 2: Related Work

There are not many attempts at analyzing data relevant to MMA. Though one attempt at it comes from the article: "Betting on UFC Fights - A Statistical Data Analysis", which uses a random forest classifier to extract several results from the data. Unfortunately, decision tree learning tends to overfit on training data, so in our case it may not be a good idea to use this method.

Regardless, even if we uncover a model that can predict accurately the outcome of a match, the model itself will not be complete since we are not incorporating the fact that previous matches for a Fighter A may influence the outcome of his/her next match, as suggested in the book: "Predictive Modeling for Sport and Gaming". As mentioned in the book, a fighter's psychology will have influence over his actions, something that we do not consider in our learning algorithms, though it would have been too difficult to do so.

For our learning algorithms, we mainly focus on logistic regression and SVMs, in which both algorithms tend to do well as mentioned in the paper: "Comparison between SVM and Logistic Regression". We found, in accordance to the paper, that both logistic regression and SVMs do similar in performance (when choosing a "good" kernel for SVMs). The paper mentions that SVMs will commonly achieve a better accuracy with less data as opposed to logistic regression, but it did not matter much in our case since we have 217 training samples (= 434 data points) to work with.

Choosing the best features is very important to SVM training. The study "Combining SVMs with Various Feature Selection Strategies" discusses using different feature selections and mapping and characterizing their effectiveness using an F-score (which measures how well these features distinguish the data points from each other). In our project, we tried different feature mapping from the fighter's attributes, which has made a difference in the results of our algorithm. If we had more resources, we could objectively score these features based on a criteria like the F-score.

In this study "Relationships between mindfulness, flow dispositions and mental skills adoption: A cluster analytic approach", the researchers used k-means clustering to group athletes into buckets according to features such as emotional control, flexibility, engagement, etc . These clusters are then compared in terms of the average performance of the athletes. This study is quite similar to our approach to classify fighters stylistically through clustering.

Section 3: Dataset

Our data was collected from FightMetric, a small company dedicated to providing data on Mixed Martial Arts (MMA) fighters and events. We collected 217 training samples and 58 testing samples from the data that FightMetric provided on the UFC matches over the past few years.

For each training sample / testing sample we collected (which is represents a match during the UFC), we do the following preprocessing: each training sample contains eight total features, in which each consist of a mapping of the following eight statistics of Fighter 1 and Fighter 2: Significant Strikes Landed per Minute, Significant Striking Accuracy, Significant Strikes Absorbed per Minute, Significant Strike Defence (the % of opponents strikes that did not land), Average Takedowns Landed per 15 minutes, Takedown Accuracy, Takedown Defense (the % of opponents TD attempts that did not land), and Average Submissions Attempted per 15 minutes.

Here, we also do something subtle: for each UFC match, we create two data points: one which consists of mapping statistics of Fighter 1 and Fighter 2 (in that order) with the outcome of the match for Fighter 1, and a second point which flips Fighter 1 with Fighter 2. This was done to prevent any possible bias in the data if we happen to list Fighter A and B as Fighter 1 and 2, respectively, or if we happen to switch them instead. Of course, this creates the assumption that the outcome of Fighter 1 in a given match is independent from the outcome of Fighter 2 in the same match, which is definitely not the case, but we regardless make this assumption for simplicity of our work.

Section 4: Methods

We used three different learning algorithms for predicting the outcome of each UFC match. They are the following: Naive Bayes classifier, Logistic Regression, Support Vector Machines (SVMs). We also implemented K-means clustering to help us categorize different styles of fighting and observe any trends between matches of different clusters.

For the Naive Bayes classifier, we sought to model the function p(x|y) where x represents the UFC match sample given and y represents whether Fighter 1 won the match (y = 1) or lost (y = 0). In the classifier, all our p(x|y)'s are estimated from the training data and then tested on the sampling data.

For Logistic Regression, we seek to determine the function: h(x) = $\frac{1}{1+e^{-\theta^T x}}$

θ here represents weights that we estimate by maximizing the following equation (the log likelihood of h(x):

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} log\, h(x^{(i)}) + (1 - y^{(i)})\, log\, (1 - h(x^{(i)}))$$

Once determining an appropriate θ that maximizes the above equation, we plug in each testing sample into our h(x) (with one edit: we add an extra attribute for each sample, $x_0$whose value is 1 (this is also done with all training samples).

For SVM, we seek to find a hyperplane that separates all sample points into two regions and which maximizes the smallest distance between a data point and the hyperplane margin. Formally, we wish to minimize the following equation:

$$min_{\gamma,w,b}\ 0.5||w||^2 + C\sum_{i=1}^{m} z_i$$

$$\text{such that } y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$$

Here, the $0.5||w||^2$ term is the distance from the data point to the separating hyperplane. Of course, most data sets in reality are not linearly separable, which explains the $C\sum_{i=1}^{m} z_i$ term; this term incorporates a penalty for when we misclassify a data point. Finally, the constraint ensures that our margin is greater than one (as opposed to less than one or even negative since we are trying to correctly classify data and not misclassify it).

The K-means clustering algorithm is an iterative unsupervised learning algorithm where k cluster centroids are randomly initialized. At each iteration, training examples are assigned to the closest centroid and each centroid is updated to be the mean of all the training examples assigned to it. This algorithm is guaranteed to yield convergence in practice.

Section 5: Experiments / Discussion / Results

We first formatted the data extracted from FightMetric in the following way: for each statistic X listed for each Fighter (1 and 2) of a match, we take the statistics X_1 and X_2 and map them to a feature X'. This is done for the following eight statistics from each fighter: Significant Strikes Landed per Minute, Significant Striking Accuracy, Significant Strikes Absorbed per Minute, Significant Strike Defense (the % of opponents strikes that did not land), Average Takedowns Landed per 15 minutes, Takedown Accuracy, Takedown Defense (the % of opponents TD attempts that did not land), and Average Submissions Attempted per 15 minutes. In total, we will have eight features to work with. As for the mapping itself, we used three different mappings and compared the results of each, which are the following mapping functions:

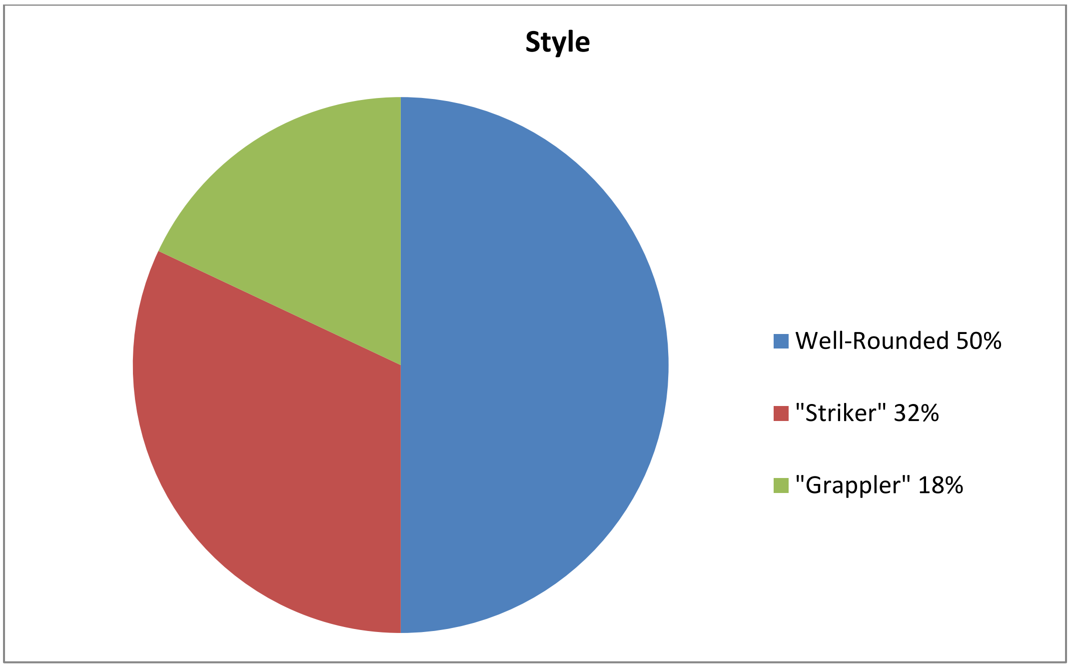
$$f_1(a_1, a_2) = a_1 / a_2,\ f_2(a_1, a_2) = a_1 - a_2, \text{ and } f_3(a_1, a_2) = (a_1 - a_2) / (a_1 + a_2)$$

Classifier	Accuracy with a_1/a_2	Accuracy with $a_1 - a_2$	Accuracy with $(a_1 - a_2)$ / $(a_1 + a_2)$
Naïve Bayes	56.89%	46.55%	53.45%
Logistic Regression	69.82%	67.24%	67.24%
Linear-kernel SVM K(u,v) = u · v + 1	68.1%	68.97%	65.52%
Normalized linear-kernel SVM K(u,v) = $\frac{1}{N}$ u · v + 1	68.97%	67.24%	64.66%
Polynomial-kernel SVM (2nd) K(u,v) = $(u \cdot v + 1)^2$	52.59%	68.97%	67.24%
Norm. polynomial-kernel SVM (2nd) K(u,v) = $(\frac{1}{N}u \cdot v + 1)^2$	54.31%	67.24%	67.24%
Polynomial-kernel SVM (3rd) K(u,v) = $(u \cdot v + 1)^3$	47.41%	51.72%	62.07%
Norm. polynomial-kernel SVM (3rd) K(u,v) = $(\frac{1}{N}u \cdot v + 1)^3$	54.31%	65.52%	68.97%
Sigmoid-kernel SVM K(u,v) = tanh(u · v + 1)	49.14%	56.9%	51.72%
Norm. sigmoid-kernel SVM K(u,v) = tanh($\frac{1}{N}$ u · v + 1)	42.24%	56.9%	64.66%
Gaussian-kernel SVM K(u,v) = $e^{(-\vert u \cdot v\vert^2)}$	64.66%	64.66%	63.79%
Norm. Gaussian-kernel SVM K(u,v) = $e^{(-\frac{1}{N}\vert u \cdot v\vert^2)}$	62.93%	66.38%	65.52%

As we can see, in general, algorithms that use f_3 as the feature mapping function do better, which makes sense since f_1 does not negate the resulting value if its arguments are flipped (while in our data, if we switch Fighter 1 with Fighter 2, the resulting classification is reversed). As for f_2, the potential problem with it is that the value it returns is not normalized, so if we double the value of the arguments, the value of $f_2$is also doubled. f_3 takes care of that for us fortunately, and also switching the arguments reverses the value returned (which is negation in our case).

For the learning algorithms used, logistic regression does best, whereas Naive Bayes does poorly and SVM does decently in general. The reason that Naive Bayes does poorly could be that not a lot of data was given to the classifier (217 * 2 = 434 training samples), and given the high dimensionality of our data for the Naive Bayes classifier, the sparse data will not allow Naive Bayes to learn well, hence its poor performance. As for SVMs, the accuracy depends partly on the kernel being applied (where linear and Gaussian seem to perform well for the most part as opposed to polynomial and sigmoid kernels). Logistic regression does well since instead of a linear dependency, it assumes a logistic dependency, which is a dependency encountered more often with big data as opposed to a linear dependency.

Clusters	SLpM	Str. Acc	SApM	Str. Def	TD Avg	TD Acc	TD Def	Sub Avg
Well-rounded	2.71	0.43	2.44	0.58	1.64	0.41	0.60	0.78
The “Striker”	4.20	0.43	3.89	0.59	0.88	0.38	0.64	0.49
The “Grappler”	3.46	0.48	2.37	0.58	4.40	0.54	0.66	1.08



“Grapplers” tend to have an advantage over Well-Rounded fighters (61% victories)
Well-Rounded fighters tend to have an advantage over “Strikers” (62% victories)
“Grapplers” and “Strikers” seem more evenly matched (53% for “Gapplers”)

Once we implemented the K-means algorithm, the decision was to choose what value of k we should use, which was based on our intuition and knowledge of mixed martial arts. We tried k = 3, 4, 5 and chose the value (k = 3) that made the most sense in clustering fighters by their styles. Observe the 3 clusters above: the well-rounded fighter, the “striker”, and the “grappler”. The “striker” is characterized by high Significant Strikes Landed per Minute (SLpM), low takedown attempts (TD Avg) and low submission attempts (Sub Avg). If the numbers could tell the story, this type of fighters likes to stay on their feet and exchange punches and kicks. The “grappler” is characterized by a different set of numbers: low SLpM, high TD Avg, and high Sub Avg. This type of fighters likes to bring the fight to the ground and avoid exchanging blows on their feet. The well-rounded fighter has medium SlpM, TD Avg, and Sub Avg, and seems to be well versed in standing and on the ground.

Assigning each fighter to a cluster allows us to observe any interesting tendencies that occur when a fighter of a certain style is matched up with a fighter of another cluster (the results are reported under the pie chart).

Fight	*Return on $100	Market Probabilities	Prediction	Prediction Probabilities
Aldo vs. McGregor	105, 80	0.43, 0.57	Aldo	0.504, 0.496
Weidman vs. Rockhold	65, 135	0.67, 0.33	Weidman	0.53, 0.47
Souza vs. Romero	67, 130	0.66, 0.34	Romero	0.49, 0.51
Maia vs. Nelson	69, 125	0.65, 0.35	Maia	0.70, 0.30
Holloway vs. Stephens	18, 425	0.96, 0.04	Holloway	0.65, 0.35
Faber vs. Saenz	13, 525	0.98, 0.02	Saenz	0.47, 0.53
Torres vs. Lybarger	36, 235	0.87, 0.13	Torres	0.95, 0.05
Alves vs. Covington	100, 83	0.45, 0.55	Alves	0.503, 0.497
Santos vs. Lee	475, 15	0.03, 0.97	Lee	0.36, 0.64
Proctor vs. Mustafaev	290, 29	0.91, 0.09	Proctor	0.73, 0.27
Makdessi vs. Medeiros	63, 140	0.69, 0.31	Makdessi	0.56, 0.44
McGee vs. Alexandre Jr.	57, 155	0.73, 0.27	McGee	0.82, 0.18

For the predictions above, we used our best-performing algorithm—logistic regression with a feature mapping function of $f_1(a_1, a_2) = a_1 / a_2$. The reported probabilities represent our values of h(x). These are compared with the market probabilities, which are computed from the betting odds the public has generated (using the inverse relationship between betting return and favorability): Market Probability(fighter 1) = Betting return on fighter 2 / (betting return on fighter 1 + betting return on fighter 2).

We can see that only in 4 of the 12 fights (highlighted in yellow) do our predictions of the winner deviate from the market. This shows that for the most part, our algorithm is consistent with the intuition of the fans and betters of the sport.

Section 6: Future Work

Some ideas for future work are including more features, making predictions on other aspects of fight outcomes, and running analysis on market predictions. For example, besides the 8 attributes we used for this project, we can build a more comprehensive profile for each fighter by incorporating height, reach, and past record or recent performance, all of which are very relevant to how a fighter is expected to perform at the next fight. We can also make prediction not just on win or loss but on the method of winning (knockout, submission, or judge’s decision) and the round in which the fighter would finish the fight. Lastly, we can extend on the market probabilities we calculated from betting odds to compare the accuracy of market probabilities with the probabilities generated from learning models.

Works Cited

Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

Chen, Yi-Wei, and Chih-Jen Lin. "Combining SVMs with Various Feature Selection Strategies." *Feature Extraction Studies in Fuzziness and Soft Computing* (2006): 315-24. Web.

Kee, Ying Hwa, and C.k. John Wang. "Relationships between Mindfulness, Flow Dispositions and Mental Skills Adoption: A Cluster Analytic Approach." *Psychology of Sport and Exercise* 9.4 (2008): 393-411. Web.

Salazar, D.A. et al. 2012. “Comparison between SVM and Logistic Regression: Which One is Better To Discriminate?” *Revista Colombiana de Estad ́ıstica*. 35, 2 (2012), 223–237

Schumaker, Robert P., Osama K. Solieman, and Hsinchun Chen. *Sports Data Mining*. New York: Springer, 2010. Print.

Singh, Vik. "Betting on UFC Fights - A Statistical Data Analysis." *Viks Blog*. N.p., 21 Sept. 2011. Web. 12 Dec. 2015.