Ruck Those Stats! Machine Learning as the New Coach

Alejandro Sanchez, Nicolas Sanchez
Stanford University
CS 229 Machine Learning, Autumn 2014

In this project we aim to identify which quantifiable aspects of the game of rugby union are most critical to carry a team to victory. To do this, we collect data from thousands of past games and use this to train several learning algorithms. Using a cross-validation algorithm we pick the most relevant features using the best available model. We then run the models again using these reduced features. Finally, we shift from looking directly at the features to analyzing a team's deviation from their past k performances in order to understand what aspects of the game usually require most focus and improvement.

1 Introduction

With the Rugby World Cup being the 4th largest sporting event in the world, Rugby is a passion for many. It is one of the most popular sports today, with established fans around the world.

By its construction, Rugby puts into motion a wide variety of skills, testing the members of a team both mentally and physically. Though much importance is usually attributed to technical skills, strength and athleticism, no experienced follower of Rugby can deny the vital strategic and team component of the game. As a result teams typically develop a strategy that prioritizes certain factors of the game over others. Some teams may focus more on ball possession while others focus on clear line breaks, or kicks from hand for territory. Understanding to what extent each of these factors contributes to a victory therefore becomes one of the main goals for a team, its coaches and the millions of fans around the world. Hence, it seems relevant to investigate what machine learning can offer in this respect.

More precisely, our aim is not to predict which team will win in a given match or given tournament. Rather by learning from which strategies have been successful in the past, we want to gain insight on the sport of Rugby as a whole, rather than focusing on a single game outcome.

2 Data Collection

We did manage to get access to readily available and easily manageable datasets to perform machine-learning algorithms. But given the popularity of Rugby Union, large collections of both team and player performance statistics are available in various websites to deliver a "Match Pack" that describes a game in detail to devoted fans.

For our project, we decided to focus on team performance statistics. The first practical reason was the high variability of the individual statistics. Since players change teams from year to year, an analysis of the individual players would probably inform us on that particular team without giving us a general insight into the sport of Rugby. Moreover, Rugby is at times referred to as the "ultimate team sport", and is without a doubt fundamentally driven by the performance of a team more than by the performance of any individual player.

We decided that the ESPNSCRUM.COM website contained the most abundant relevant team data. Our first objective was therefore to get the data of all the games (each being on a separate webpage) into one same document. This proved particularly challenging as different statistics were recorded for different matches, depending on the year, and the importance of the match. Effectively scraping useful data off of the website therefore required an elaborate python script.

Once we found a way to scrape the data, we chose to use games from Super Rugby and the English Premiership from the past 7 seasons. Both are professional league competitions for top clubs in their respective regions. Super Rugby brings together top teams from Australia, New Zealand and South Africa, while English Premiership gathers top teams from around England. Both are popular leagues, in turn meaning that ESPNSCRUM kept track of most of the statistics for the

majority of the games in the last 6 seasons. They are both larger than most other tournaments, meaning that each season had more games, leading to more training points our algorithm could learn from. We collected a total 2700 data point with ground truth.

3 Feature Selection/Preprocessing

From the website we obtained 22 different statistics of each game. Most of these were in fact multiple performance metrics in one (Rucks won/ Rucks Lost was just listed as one on the website). We preprocessed this data so as to write all out statistics and eliminated redundant statistics, as well as performance metrics that were too sparse. We therefore ended up with a staggering 38-team performance measures per game. We chose these performance metrics, as they are what both fans and coaches analyze to understand if they had a good performance or a poor performance. Indeed we obtained "game stats" sheets from a former professional rugby player, and the stats they analyzed to evaluate their game performance were similar to the list of performance metrics we included in our metrics.

In line with our objective, rather than comparing the performances of opposing teams, we only considered the performances of one team as the input, and classified it depending on whether the given team won the given match. Initially we used all of these features for our first algorithm. Seen as the ranges of the performance metrics varied greatly (a team typically runs hundreds of meters while they only get three or four tries per game) we decided to standardize the features (mean removal and variance scaling) to receive aid our models.

We began our project by having three distinct classes for wins, losses and draws. As soon as we started to look at the data and the result of the first machine learning algorithm we realized that there were only a small number of draws (a little less 3% of the games were draws) and their presences significantly dropped the accuracy of our models. We therefore decided to rid both our training set and test set of all draws. Again we do not believe this goes against our objective since our mission is to gain insight on the game of Rugby and we consider gaining insight on 97% of the game will probably generalize to that last 3%.

4 Two Approaches

We took two different approaches to creating the features we would input to a machine-learning algorithm.

Our first approach was to use the list of performance metrics for a given game for a given team as the features for a training input and classifying that training point as won if the given team won the given game and as a loss if the given team lost the given game. We train the different models described below to this training data. Intuitively, this first approach takes games on an individual basis, looks at the team's performance, and tries to model how each team has to perform with respect to each feature in order to win or lose a game. If we have a strong model that accurately predicts if a team won game given its performance, analyzing which features are most important for the model should give us an idea of which performance metrics are most important for a team to win.

Our second approach is a little more elaborate and comes from another strong professional belief in Rugby that to do well in a game you simply have to have certain performances metrics better than your usual performance. For each performance metric we look at the performance of the team in the game that we are analyzing, and subtract it from the same teams previous k games. The vector containing these values for all of the features will be our input vector for the game. We still classify this input as we did in the first input ("win" if the given team wins the given game). Intuitively this approach should have more information on how the team is performing with respect to its own standard in recent games, and might therefore give a better idea if a team played well or not.

5 Model Selection and Results

Before looking at reducing our features to an optimal set, we needed to select a model that could best predict the outputs of a game using the given team statistics. The models we chose to explore were, Naïve Bayes, SVM, Random Forest and Nearest Centroid. We ran a first iteration of each of these algorithms using the raw form of the data, where we only made necessary adjustments to be able to run the algorithms. We then made a series of transformations to our data in order to increase the success rate of our algorithms, including eliminating draws, normalizing the data and using k-past season performances for each team.

5.1 Initial run, setting the baseline

As we explored the possibilities of proceeding with this project it was necessary that we have a model that would give us basic results and more importantly a realistic baseline. With this in mind we used the Naïve Bayes algorithm. With no prior knowledge on the distribution of the features, we guessed a Gaussian distribution and ran the corresponding Naïve Bayes algorithm[1] with:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

We inputted the raw feature matrix with the corresponding label vector and received a score on both the training and test set. The Gaussian model reported a success rate of 16.7% on the training set and 24.2% on the test set. These low results were largely due to the fact that we were considering wins, draws and losses. But when we realized that only 3% of games were draws, we took these out of the modeling. Running the new data through the Gaussian Naïve Bayes, we got a success rate of 77.9% on training data and 71.3 on test data.

5.2 First Approach

Given the above results, we decided to proceed using the data without draws and preprocessing such that it was standardized to a Gaussian distribution of mean 0.

	No Draws(Standardized)		Reduced Features	
	Train	Test	Train	Test
Naïve Bayes Gaussian	0.779	0.713	0.775	0.74
SVM Linear	0.901	0.775	0.841	0.785
SVC Polynomial	0.91	0.753	0.899	0.764
SVC RBF	0.901	0.775	0.89	0.77
Nearest Centroid	0.793	0.752	0.788	0.754
Random Forest	0.988	0.72	0.993	0.723

Table 1 – Percentage of successful predictions for all algorithms using initial features and reduced features on both training and testing sets.

However, one of the characteristics of the Naïve Bayes is that it naively assumes independence on the features, which is not something we necessarily want to assume which is why we wanted to use another approach that would not make this assumption.

The next model we looked at, was the SVM. For this we used a Support Vector Classifier[1] with linear, polynomial, and RBF kernels and the results can be seen in *Table 1*. The results barely surpassed those from the Naïve Bayes with a maximum success rate form the linear Kernel at 90.1% on the train set and 77.5% on the test set. Although this is much better than a 50/50 guess, it is still not much better than our baseline.

We also used a Nearest Centroid[1], where the prediction was based on how similar a team's performance was to another game that had already been played. This gave us a success rate of 79.3% on the training set and a 75.2% on the test set.

Finally, although not something that was covered in class, we decided to use a Random Forest[1]. Using the same data that we inputted for the SVMs, we received success rates of 98.8% on the training set and 72% on the test set.

5.3 Feature Ranking and Feature Selection

The next step is to understand how relevant are the features we used. For this did an individual feature ranking using an extra-trees classifier[2]. From *Figure* 1 we notice that the most important features were tries and conversions made while red cards and mauls won are not all that relevant.

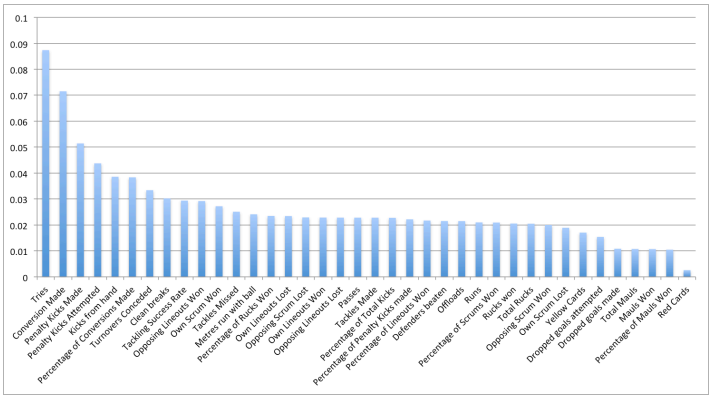

Figure 1 – Ranking of initial 38 features using an extra-trees classifier. Tries and Conversions made rank at the top while Percentage of Mauls won and Red Cards lag at the bottom.

Given this information we looked to reduce our feature set in order to get a better sense on what are the core features that define a successful rugby game. Hence we carried out a Recursive Feature Selection[1] using a linear SVM, and *Figure 2* shows that the optimal number of features was 23. We note that with the first 5 features there was a rapid increase in success rate but then the rate stagnates with only an overall slight increase.

With these most relevant features, we trimmed our data to reflect this change and plugged in the new data into the algorithms we previously mentioned. *Figure 1* contains the results for this reduced data. It can be noted that there was an overall increase but it was not considerable (Naïve Bayes had the highest increase with 2%) An interesting point is that this feature selection had an increase on all models even if the selection was done using a linear SVM.

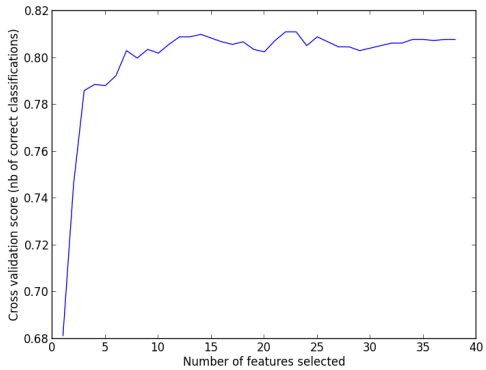

Figure 2 – Graph of cross validation score given the number of features selected. The maximum is at 23 features with a score of 0.818. Note that this is a cross-validation score and not a score on training or testing sets.

5.4 Second Approach

As mentioned earlier, this approach seeks to take into account the momentum of a season, and model what aspects of the game, or in this case features, that need to be improved from game to game in order to increase the chances of winning the next game.

We modified the data as described in part 4 and inputted it once more to all of our algorithms and created 6 new data sets. In each we respectively took the averages of the past 1, 5, 7, 10, 12, and 14 performances. We saw a slight but clear increase in success rate from 1 to 10 past performances but it then declined as we considered 12 and 14. In *Table 2* reports the results for the past 5 and 14 performances. And *Table 3* reports the results for the past 10 performances.

	5-Past Performances		14-Past Performances	
	Train	Test	Train	Test
Naïve Bayes Gaussian	0.721	0.712	0.737	0.704
SVM Linear	0.759	0.738	0.813	0.709
SVC Polynomial	0.91	0.711	0.96	0.69
SVC RBF	0.865	0.72	0.906	0.709
Nearest Centroid	0.732	0.704	0.76	0.682
Random Forest	0.99	0.694	0.993	0.668

Table 2 – Models run using data with averages of past 5 and past 14 performances.

	10-Past Performances		10-Past Reduced	
	Train	Test	Train	Test
Naïve Bayes Gaussian	0.706	0.718	0.699	0.716
SVM Linear	0.759	0.791	0.765	0.799
SVC Polynomial	0.921	0.762	0.887	0.756
SVC RBF	0.868	0.7742	0.859	0.781
Nearest Centroid	0.715	0.733	0.719	0.75
Random Forest	0.99	0.721	0.984	0.704

Table 3 –Models run using past 10 performances. This was the model yielding best results.

As we did in the first approach we now want to know what the ranking of the individual features is in order to get a grasp of what were the most relevant factors in deciding whether a team wins or loses the game. *Figure 3* shows the ranking.

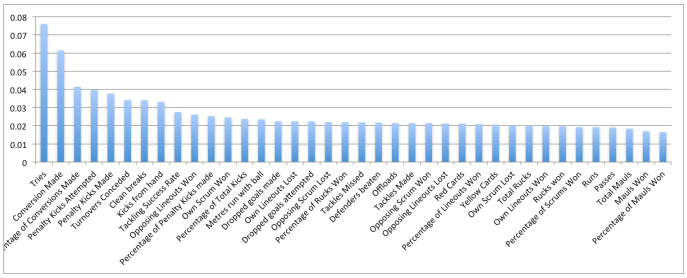

Figure 3 – Ranking of 38 features using an extra-trees classifier. Tries and Conversions made rank at the top while Percentage of Mauls won and Red Cards lag at the bottom. Due to an issue of scaling not all features are labeled above.

Finally we want to see which features were the most relevant and if we could settle for a set of core features to get just as accurate or more accurate predictions than we did with all 38 features. *Figure 4* shows the result of the RFS ran on the data with past 10 performances.

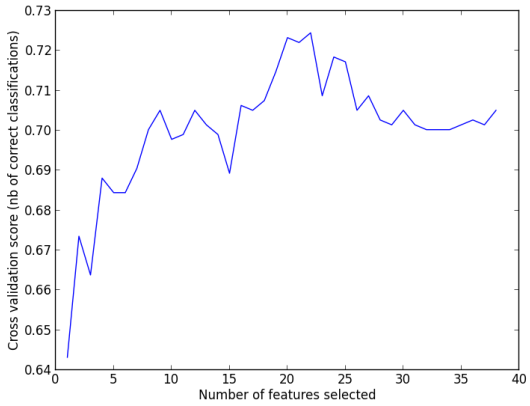

Figure 4 – Graph of cross validation score given the number of features selected. The maximum is at 22 features with a score of 0.723. Note that this is a cross-validation score and not a score on training or testing sets.

Table 3 reflects the change when we run the trimmed data on the models once more. We observe that the linear SVM yields the best results with 79.9% accuracy on the test set.

6 Discussion

6.1 First Approach

For the first approach we see that our individual rankings graph strongly indicates that there are a few rankings that are significantly more important than the rest. Upon inspection, we see that this corresponds to the point scoring performance metrics such as tries scored, conversions made and penalty kicks made. Since the winner of a game is ultimately determined by the team that has scored the most points, it make sense that these features are most important in evaluating whether a team has most likely won the game or not.

We notice however that the other factors are in no way useless. Indeed, the model that best predicts the outcome of the game based on the performance metrics of the team in question takes into account 23 features. So although point scoring metrics are the most important, a number of other metrics still bring valuable information about the team's likelihood of winning that goes beyond the information given by the point scoring metrics. This goes in line with the general idea that to win a game, one should focus on some important factors of the game apart from scoring tries.

In order to better understand these other non-scoring metric features, we reduced the original data with 38 features. This time, we eliminated the 5 scoring metric features, which were: Tries, conversions made, percentage of conversions made, penalty kicks attempted, penalty kicks made. We ranked the features individually one more time, and the results can be seen in *Figure 5*.

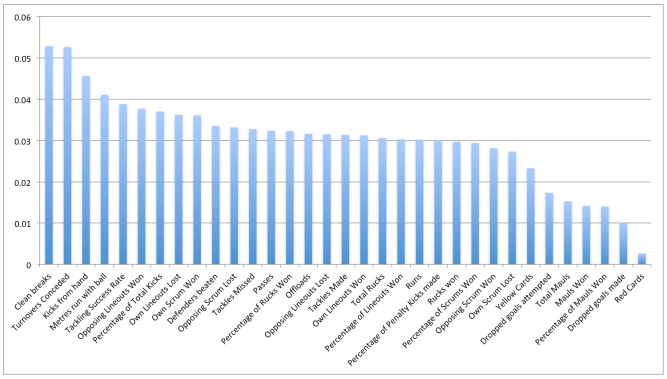

Figure 5 – Ranking of non-scoring metric features using an extra-trees classifier. We see a different ranking than what we saw in the Figure 1.

6.2 Second Approach

With the second approach, we see that the model is most accurate when trained with 10 past performances. However this may also have to do with the fact that a team has no more than 16 games in a season and hence higher values of k lead to much less data points both for training and testing.

An initial feature ranking shows that once again in general a team should always try to score more points than it has been doing so far, whether it is by making penalty kicks or scoring tries. This remains significantly more important than all the other performance metrics. However, finding the optimal combination of metrics yields an optimal number of 22 performance metrics, which is again much higher than the 5 or 6 highest individually ranked performance metrics. This time requiring more features is even more important since the cross validation scores drop notably for less than optimal performance metrics.

7 Conclusion

With 10 past games, the model is more accurate than with the first approach, both with all of the features and the reduced features.
By construction of our two methods this suggests that it is more accurate to evaluate how well a team played by considering their recent past performances than thinking there is some absolute performance formula for a well-played game of Rugby. Nevertheless, they both confirm that to be successful, a team must excel in a variety of performance metrics, which most likely lead to success only when they are combined. This optimal combination of features seems to be somewhat stable as the 22 chosen optimal features using 10 past games mostly overlaps with the 23 features chosen for the first approach. Furthermore these performance metrics are in accordance with the performance metrics professionals used to analyze their games.

8 Future

Our next step would be to see if our results still hold when we test them on games from other tournaments, and see how that impacts the accuracy the models and hence the importance of our chosen features.

Even though we deliberately took a different approach, it would also be interesting to see what the results would be like if we combined the performances of opposing teams. Obviously leaving point scoring metrics, we could evaluate at which metric should a team outperform the opposing team.

9 References

[1]Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011

[2] P. Geurts, D. Ernst., and L. Wehenkel, "Extremely randomized trees", Machine Learning, 63(1), 3-42, 2006.

Patel, S. Parity and Predictability in the National Football League
URL http://cs229.stanford.edu/proj2013/Patel-ParityAndPredictabilityInTheNationalFootballLeague.pdf