Study of KPIs which affect the outcome in Rugby Union fixtures

Sokratis-Dimitrios Chronopoulos (Student)

Monday, 25th March 2019

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank Rob Holdsworth on behalf of Scottish Rugby Union Performance Division for providing me with the project details and the database and helping me understand several rugby terms.

Also, I would like to thank all the academic and administrative personnel with I interacted during my master's year. I need to express my gratitude especially to the master's programme directors, Professor Kerem Akartunali and Dominic Finn, alongside my academic supervisor, Professor Viktor Dörfler, for the empathy they felt for me and the support that they provided me during the project/dissertation period so that I manage to overcome all the challenges I was facing during this period. Without their invaluable support, I could not succeed to complete my dissertation.

Moreover, I would like to thank my new friends and classmates from the University of Strathclyde especially Georgios, Giannis and Konstantinos for their genuine care and support on my efforts during the master's period. Beyond them, Despoina, Kyriakos, Constantinos, Vasilis were always there for me to hear my challenges and encourage me believing in my potentials.

Last but not least, I want to thank my family for their sacrifices during my -academic and not only- life and the way in which they have raised me. Without them, I would not be the person I am. Above all, I want to praise the One True God, the Holy Trinity for all the goods, the facilitation, the faith to myself and the blessings I have received during my life.

Executive Summary

The challenge for SRU performance analysts is to manually analyse a complex set of events happening during a match and determine which events affect the game outcome. The aim of the conjoint project between SRU and the Strathclyde Business School was the development of an effective and efficient computer-aided solution that will provide the Scottish Rugby Union performance analysts with the most updated KPIs which will bring an effect on winnings/losses. This solution should harness the potentials of Machine Learning techniques.

After retrieving data for 323 games plus a validation sample of the most successful home teams, two classifiers were constructed. After assessing these two classifiers, Classifier 2 selected. The selected classifier's Accuracy was 75.8% in average. Twenty-two performance indicators (out of twenty-seven that the model includes) compose ten groups associated with game actions. These performance indicators were divided into two categories according to whether to reinforce or limit the odds of winning. The KPIs that increase the log(odds) of winning stemmed from actions that help teams maintain the possession, gain meters/territory and utilise individual player's skills (power, shot/pass accuracy, velocity) in order to score tries and goal kicks mainly. On the other hand, teams that fail to defend effectively or keep possession and make errors on the transition game from their side to the opponent's has fewer opportunities to win a match. Referee's intervention seems positive to the game outcome for a team, but it is not clear which kind of intervention is which helps.

This classifier was considered as competitive among previous implementations. Points of development that were noticed are related to the ways of handling Not Available (NA) values and training a model and the future implementation of other Machine Learning models. Moreover, more engagement of the performance analysts could be beneficial for constructing a robust model.

Introduction

In all sports -either team or individual- the primary goal is to achieve a winning outcome. The outcome is affected by events taking place throughout the game duration. There are sports where games end after a specific time and others where games only end after a team or an individual reaches a winning score predetermined by the rules of the sport. The combination and variation of Performance Indicators (PIs) are factors that make the investigation of win covariates a complex task.

Speaking about team sports -among them Rugby-, there was an initial period where each sport lacked professionalism. While the people participating in sports were predominantly amateurs, there was no real need for professional game analysis. Empirical post-match analysis conducted by the coaches was considered adequate in order to find the strengths and the weaknesses of the team and to construct the strategy and tactics for upcoming fixtures. Coaches used to believe that they were capable of extracting conclusions on the critical elements of game performance without any observational aid. However, various studies have proved that not only coaches fail to recall more than a middle-low 45-59% of events occurring during a match ((Franks & Miller, 1991) and (Laird & Waters, 2008)) but also tend to misjudge differences in performance and to identify actual differences at the same level of inexperienced/novice coaches as well (Franks, 1993).

From the time since Rugby transformed to a professional sport, teams have been oriented towards investing resources on developing methods and tools of game analysis which will provide them with accurate and sufficient key performance indicators (KPIs) that affect the game outcome. The evolution of technology allowed the recording of the athletic events by cameras for TV-broadcasting. Team staff leveraging this service started to manually capture movements by inspection of video recordings from past matches. These data were used to produce descriptive statistics to support movement and tactical analyses. This fundamental shift has enabled performance analysts to be hired in order to fulfil this role.

Nowadays, automatic movement recording is provided by several data vendors, such as OPTA and STATS, in the form of services capable of extracting movement, event, and statistical data based on their own recordings (Stein, et al., 2017). Along with technology evolution, computer scientists developed new algorithms capable of solving complex mathematical models by harnessing the power of personal computers. Several sophisticated algorithms from Artificial Intelligence (AI) are now employed in Sports Data Analysis.

Rugby Associations and professional teams have been aware of the intervention of Sports Analytics in Performance Analysis. So, they take advantage of the potentials of Sports Analytics by hiring performance analysts and data scientists to analyse the available data, identify trends, measure the

team/player performance and provide coaches with actionable insights; in other words, the obtained scientific staff enable the coaching staff to make better data-driven decisions.

In this context, Scottish Rugby Union (SRU) has created the Performance Division to run everything related to performance for a variety of teams associated with Scottish Rugby Union, such as the local rugby teams and the National Team.

Project Definition

The Performance Division of Scottish Rugby Union (from now on, SRU Performance Division) aims to provide teams with insight obtained through post-match data analysis contributing to the ability of the coaching staff to rival team scouting and develop its own team strategy adaptation according to the needs of forthcoming fixtures. For doing so, the SRU Performance Division has adopted video analysis technology provided by OPTA and maintains a database with all fixtures taken place every year covering 25 competitions over the world.

The challenge for the SRU performance analysts is to explore the depth of measure that OPTA software captures and records in a database. That means that the analysts must manually explore a wide range of PIs to assess and determine which of them must be characterised as KPIs. This process demands a significant amount of time because of the vast magnitude of the datasets. Except for the -high- time complexity that such a process requires to run, there is also the risk of analysts omitting factors critical for the game outcome -due to the urge for quick analysis- or including factors that more explain each other than the response variable, namely the game outcome; that is denoted the phenomenon of collinearity (Belsley, et al., 1980). Moreover, the continuous change of the Rugby laws affects the game-play and thus the significance of PIs. The analysts have slowly come to recognise this effect.

The arisen opportunity is that there have been several studies for several team sports on KPIs influencing match outcome in which sophisticated methods have been used to create classification models predicting the game outcome. The results were very encouraging as the models could estimate the game outcome at a high accuracy level. These models are created by data scientists, professionals with high-level programming skills, who are experienced in data mining from large databases and possess applied knowledge in statistical data analysis and machine learning.

The purpose of this project is the development of an effective and efficient computer-aided solution that will provide the Scottish Rugby Union performance analysts with the most updated KPIs which will bring an effect on winnings/losses. The project constitutes a partnership between the SRU and the Strathclyde Business School.

The questions that the solution must provide are:

1. Is there any Machine Learning implementation capable of predicting the game outcome efficiently?
2. Which are the KPIs that affect the game outcome in Rugby union games?

The conjoint project has been assigned to the researcher through the dissertation selection process. The researcher is requested to expose a combination of scientific (research) and technical (data science) skills. To achieve the primary purpose, the researcher has set the following milestones:

- ☑ Do individual research on methods applied to relevant subjects, i.e. previous studies on KPIs affecting the game outcome in Rugby and other sports.
- ☑ Pre-process secondary data, transform them and use them to create classifiers based on Machine Learning (ML) models (R-script executables manipulating data from the PowerBI database)
- ☑ Evaluate the classifiers and compare these implementations to other approaches on extracting KPIs in Rugby.
- ☑ Present the KPIs as they have been generated from the implemented solution. Compare these KPIs to others produced from other studies.

Stakeholders of the project are:

- Robert Holdsworth, Lead Performance Analyst (SRU Performance Division) – contact person
- Sokratis-Dimitrios Chronopoulos, MSc Student in Data Analytics (University of Strathclyde) – Data Scientist Intern at SRU
- Dr Viktor Dörfler, Professor (University of Strathclyde) – Academic Supervisor

Scope of the Project

The deliverable of this project is expected to provide the SRU Performance Division stuff with the most significant game outcome factors. These factors would be used in order to enable coaches to create new attacking and defending tactics according to the special characteristics of both their own and the opposing team. As the game season progresses, new post-game data would feed the already created model adjusting the KPIs' influence and subsequently providing up-to-date information on how the teams' strategy changes. Moreover, as the SRU aims to invest in growing Rugby in Scotland and upgrading the level of the existing teams, this deliverable would be used for providing customised consulting to teams in order to better understand their points of development.

Methodology

The nature of the study

Analysing the outcome of a Rugby match, three possible events can happen:

1. Win for Home Team / Loss for Away Team
2. Loss for Home Team / Win for Away Team
3. Draw between the Home Team and the Away Team

A draw is less likely to be noted as a result in rugby union games due to the variety of different ways to score and different game points that these ways yield. For this reason, the draw results have been set out of the scope of this research.

The actions that occur during matches are recorded and stored in a database. These actions synthesise the match outcome. Modelling this relationship between fixture result and actions occurred during this fixture, the abstract model describing this relationship is the following:

$$\text{Fixture Result} \sim \text{fixture}(\text{Actions}),$$

where the Fixture Result is the response (dependent) variable, Actions are the set of the explanatory (independent) variables and ~ the symbol expressing the existence of a dependence.

Not all of them have the same value as regards to the impact that they make on rugby game outcome. However, it is estimated that only a short branch of variables (actions) may explain the majority of rugby fixture results.

Classification

As the fixture result is a nominal variable, the problem is nominated as a classification problem (Wikipedia contributors, 2019). In Machine Learning and Statistics, classification is a problem of identifying the category -from a set of categories- to which a new observation belongs based on an obtained dataset containing instances whose class membership is already known. To resolve this problem, a classifier or a classification rule must be built.

Initially, the fixture result of a team was considered that would probably be assigned to one of the three possible classes. By omitting the draw result from this study, two possible values remain; thus, the problem can be specified as a *binary (or binomial) classification problem*.

Classifier

A classifier is an algorithm that maps input data to a category. This classifier is based on a produced function that takes the observations of a population as input and gives the class of these observations as output (Wikipedia contributors, 2018).

In order to construct an accurate classifier, a sample dataset (of the population) with the classes of the observations already known is used as a training dataset. The target of the classification learning process is to combine the features (explanatory variables) of the observations which can achieve a better successful classification rate not only for the existing dataset but also for future observations (author's note: more details on training procedure are given below in section Project Lifecycle).

Evaluation of the classifier

The evaluation of the classifier is conducted after running a classification test which is the application of the classification rule (or classifier) to a sample of the initial dataset. As a consequence of using the results of the classification test, a confusion matrix is created. A confusion matrix is, in fact, a contingency table of the frequencies of combinations of the actual and the predicted classes of the response variable, namely of the game outcome. For the purposes of this study, let Win the positive game outcome for a team and Loss the negative one.

Popular metrics derived from this matrix used for evaluating the model performance are accuracy, specificity, sensitivity and possible products of them. The notation of the entries of the confusion matrix are as follows:

- True Positive (TP) is the number of correct predictions that the actual outcome is Win.
- True Negative (TN) is the number of correct predictions that the actual outcome is Loss
- False Positive (FP) is the number of incorrect predictions that the actual outcome is Win
- False Negative (FN) is the number of incorrect predictions that the actual outcome is Loss

The confusion matrix for the binary classification problem is given in *Figure 1*.

Actual	Predicted		
		Positive	Negative
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 1 The confusion matrix for the binary classification problem (Analytics Vidhya, 2016)

By obtaining these counts from the confusion matrix, the abovementioned metrics are produced. In particular, the accuracy (AC) is the proportion of the total number of correct predictions and is calculated by:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

Another metric is the true positive rate (TPR), otherwise known as Sensitivity. Sensitivity gives a measure of how well the model can predict the actual positive outcomes. It is defined as the proportion of positive cases that were correctly identified, calculated by:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

Respectively, the true negative rate (TNR) otherwise known as Specificity gives a measure of how well the model can predict the actual negative outcomes. It is defined as the proportion of negative cases that were correctly identified, calculated by:

$$\text{Specificity} = \frac{TN}{TN+FP}$$

In contrast to the TNR, there is also the false positive rate (FPR). The FPR gives the supplement of the specificity and explains the rate of failure of the classifier to correctly predict the actual negative outcomes. FPR is calculated by:

$$\text{False Positive Rate} = 1 - \text{Specificity}$$

In cases where the two classes are not equally represented into the dataset, the balanced accuracy is taken into account instead of the typical accuracy. The balanced accuracy is calculated by the average of the true positive and negative rates (Fletcher & Fletcher, 2005):

$$\text{Balanced Accuracy} = \frac{1}{2}\left[Sensitivity + Specificity\right] = \frac{1}{2}\left[\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right]$$

Moreover, the proportion of occurrence of the positive outcome into the sample is captured. This measure is called prevalence and is calculated by:

$$\text{Prevalence} = \frac{\text{FN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

Another measure used in evaluating and comparing models is Cohen's Kappa (Landis & Koch, 1977). This is essentially a measure of how well the classifier performed as compared to how well it would have performed merely by chance (random classifier). Let p_o the observed agreement and p_e the expected agreement by chance. Then, Cohen's kappa is calculated by:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

As p_0 is equal to Accuracy and the p_e is set to 0.5, then:

$$\kappa = \frac{Accuracy - 0.5}{0.5} = 2 \times Accuracy - 1$$

The Kappa values and its interpretation of the performance of the classifier are given in *Table 1*.

Kappa(κ)	Accuracy	Performance
0.81 – 1	>0.90-1	Almost Perfect
0.61 – 0.80	>0.80-0.90	Substantial
0.41 – 0.60	>0.70-0.80	Moderate
0.21 – 0.40	>0.60-0.7	Fair
0.00 – 0.20	0.50-0.60	Slight

Table 1 Model performance in terms of Cohen's κ associated with Accuracy (based on Landis & Koch (1977))

Except for the numerical performance measurement of the model, there is an alternative empirical way of evaluating a model. This is through studying the Receiver Operating Characteristic (ROC) graph (*Figure 2*). In this plot, the Sensitivity is set against the FPR. The ideal classifier would "yield a point in

the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity and 100% specificity. The (0,1) point is also called a perfect classification. A completely random guess would give a point along a diagonal line from the left bottom to the top right" (Kumar, 2013). The area under the curve AUC) implies the measure of performance of the classifier.

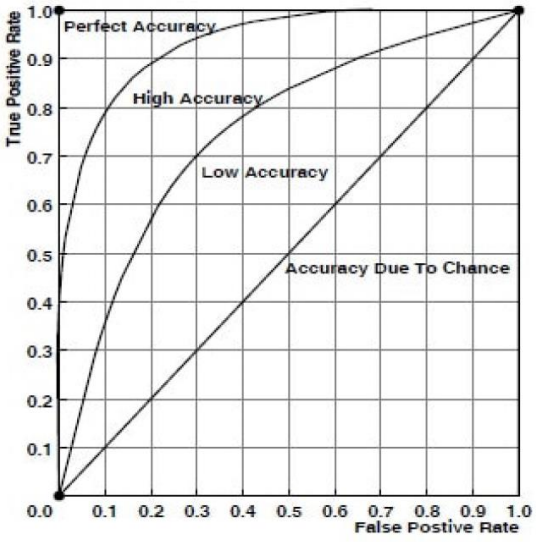

Figure 2 General form of a ROC curve

Previous studies

Original studies used statistical methods applied to small data-sets (samples with n<=100) and with a short list of PIs analysed for statistical significance upon the outcome ((Jones, et al., 2004); (Ortega, et al., 2009); (Bishop & Barnes, 2013)). Subsequent studies leveraged more sophisticated techniques originated from Machine Learning to create predictive models for the outcome ((Vaz, et al., 2011), (Woods, et al., 2017), (Parmar, et al., 2018a), (Parmar, et al., 2018b)) noticing exceptional performance (65-90%). These studies examined wider samples (100<n<600). Literature justifies the evolution of performance analysis from descriptive analytics to predictive analytics for two main reasons: 1) the explosion of the data volume in these days and 2) the eager for more in-depth observation/record and analysis of the Performance Indicators around the Rugby games.

Something else which has been observed is that the studies focused on one or at most two Rugby Tournaments each ((Ortega, et al., 2009), (Vaz, et al., 2011), (Higham, et al., 2014), (Hughes, et al., 2017), (Woods, et al., 2017), (Parmar, et al., 2018a), (Parmar, et al., 2018b)) or one team (Jones, et al., 2004) or a subset of specified teams (van Rooyen et al. (2006); Van den Berg and Malan (2010); Wheeler et al. (2010); Vaz et al. (2015), as cited in (Watson, et al., 2017)). Watson and colleagues (2017) also found that "some authors have focused on close games (Vaz, Van Rooyen, & Sampaio, 2010) and close and balanced games (Vaz, Mouchet, Carreras, & Morente, 2011) in RWCs, Super

Rugby, the Six Nations and other international games". Finally, several studies have been conducted on different aspects of game-play ((Vaz, et al., 2011), (Bishop & Barnes, 2013), (Higham, et al., 2014), (Hughes, et al., 2017), as cited in (Watson, et al., 2017)) and close matches (Vaz, et al., 2011).

Jones and partners (2004) concluded that the outcomes of the games of a team during a European domestic league were affected by the number of winning opposition lineouts and tries scored. In a study of Six Nations games between 2003-2006, Ortega and partners (2009) found that winning teams were successful because of the points they scored, especially the points that came from conversion and successful drop goal attempts, won mauls, turnovers, line brakes, possessions kicked and tackles the teams completed. Another statistical study between the International Rugby Board (IRB) competitions (World Cup and Six Nations) and Super Twelve Tournament (S12) (Vaz, et al., 2011) showed that possession kicked, tackles made, passes completed, rucks and pass, mauls won, kicks to touch and errors made affected the close games (reclassification success rate: 78%) while tackles missed, lineouts lost, possession kicked and turnovers won affected the balanced games (reclassification success rate: 73%).

Other studies considered in aggregate the pitch position of penalties conceded and the total numbers of kicks out of hand (2011 Rugby World Cup, (Bishop & Barnes, 2013)), the ball possession, the passing, the count of rucks, mauls, turnovers, penalties free kicks (2011 & 2012 International Rugby Board Sevens World Series, (Higham, et al., 2014)) and possession in the opposition 22–50, attack with ball in hand following a kick along with lineouts won on the opposition ball as the KPIs affecting the games outcomes.

In contrast with the majority of studies in which the researchers used only statistical tools and tests, the advanced studies have been evaluated in term of accuracy. Focusing on studies that have benefited the use of Machine Learning techniques, the results seem to be very encouraging. Woods and partners (2017) studied the outcomes of the 2016 National Rugby League games creating a Conditional Interference Classification Tree. The results showed that five performance indicators ('try assists', 'all run meters', 'offloads', 'line breaks' and 'dummy half runs') were retained within the classification tree, detecting 66% of the losses (specificity) and 91% of the wins (sensitivity).

Finally, Parmar and partners (2018a) analysed games from all 27 rounds of the 2012, 2013 and 2014 European Super League seasons. They applied Principal Component Analysis (PCA) to 46 features and created 10 principal components which explained the 73.4 % of sample variance and were grouped into four main categories: possession, speed of play, form and infringements. After that, they used these principal components as the input to a logistic regression classifier and an Exhaustive CHAID decision tree. The two machine learning models performed to correctly classify the game outcome at 90% and 78% of the observations respectively.

Project Life Cycle

In this section, the steps which were followed, and the methods used for the project implementation are presented. The project life cycle contains four main procedures (*Figure 3*):

1) The collection and the pre-processing of data
2) The analysis of the data
3) The implementation of ML models which lead to answer the project questions
4) The discussion of the results and the deployment of the model in SRU Performance Division tasks

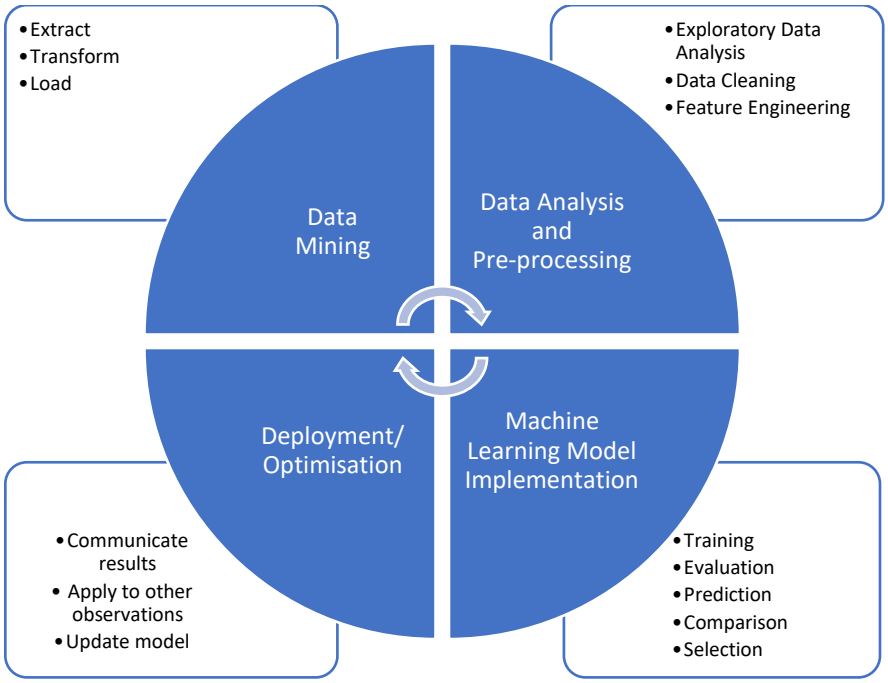

Figure 3 Project Life Cycle

Data Mining

For the purposes of this project, the SRU Performance Division gave access to real post-match data. These data have been recorded and stored in a relational database from OPTA (Opta Sports, London, UK) and have been loaded onto Microsoft Power BI Platform (Microsoft Corporation, Redmond, Washington, USA). The original form of the data permits the versatile production of queries and application of filters that match several criteria. The data mining procedure followed the Extract-Transform-Load (ETL) principles (Zhao, 2017). The data has been analysed using R language (Edition

3.5.1) using the well-known integrated development environment R-Studio (RStudio, Inc.), the written scripts will be attached to a Power BI R-functionality. This means that when someone selects data to extract from Power BI and decides to open R-Studio for running the scripts, the data will be automatically loaded in a new R-session (as a data frame) along with the code. Quick execution of the code is made feasible without any changes needed to be performed.

Data Pre-processing and Analysis

In this stage, a preliminary variable validation and selection are performed. A technique to perform this process is Exploratory Data Analysis (EDA). EDA is more an empirical and intuitive technique rather than a scientific one as the statistical tests or criteria that are used for model selection. Another property of EDA is the ability for assessing whether the given variables need transformation or re-expression. Variable types are recognised (e.g. numeric, categorical, etc.) and variables are checked for the existence of Not Available (NA) values. Where NA existence is observed, a thumb rule of dropping the variable that shows the existence of NA at more of 40% of the observations is applied.

After the initial drop of problematic variables, the dataset is being reshaped/transformed according to the selected ML technique and its unique properties. For the purposes of the study, the categorical variables were transformed into numerical. This transformation was achieved by generating all the possible combinations of the levels of the categorical variables (counts). Again, an investigation of the occurrence of NA values and redundant columns is performed, and the rule of 40% is applied as well.

When say seeking redundant columns it means that due to the previous transformation some columns may share identical data. This may happen because one level of the last merged variable corresponds to NA value, so there are two identical variables. For example, let say there are three variables: X1, X2, X3 and levels LX1, LX2, NA are levels of X1, X2, X3 respectively. So, when creating the combinations, the generated variables are LX1_LX2 and LX1_LX2_NA (assuming that the generation of variables is a sequential process using each generated variable as the first parent and the level of the next variable as the second parent). In this case, the NA level does not add any value to the count of combinations (NA is equal to non-existence). So, LX1_LX2 and LX1_LX2_NA are identical and the second one should be dropped.

Finally, it is possible that any variable that remains after the application of the rule of 40% has NA values. For this reason, the researcher decided to heal the variables with NA values populating them with the modes of the respective variables. The mode is the most frequent value observed in a variable. Where mode is used, the hypothesis is that if someone wanted to guess the value of a variable for a random game, a good bet would be the mode (Statistics Canada, 2017). If there is more than one mode, then the rounded averaged is calculated and used. Mode selected as the appropriate indicator towards

others such as the mean or the median because the variables' values indicate the frequency of occurrence of the original categorical variables. Generally, the usual approach for categorical variables is to fulfil cells with NA values with the mode of the variable.

Machine Learning Model Implementation

Machine Learning uses a data modelling approach. The Data Modelling Process has three significant steps (*Figure 4 Data Modelling Cycle* :

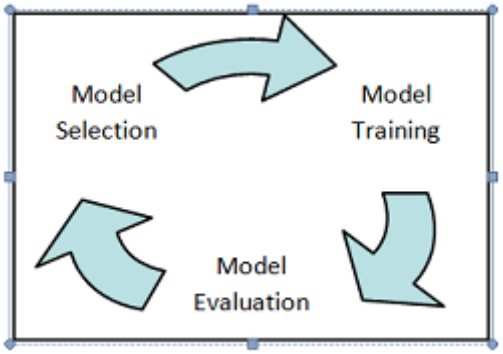

Figure 4 Data Modelling Cycle (Gorakala, 2017)

- **Step 1-Model Selection**: The model that is going to be implemented depends on the goal of the analysis is performed. For example, if the issue is a classification, then a Supervised Machine Learning Model is recommended (Shetty, 2018).
- **Step 2-Model Training**: After choosing the model, the prepared data is going to be split into train and test sets. In Supervised Machine Learning, the training data is used to train the model on how to predict the game outcome. Afterwards, this model uses test data to fit the data into the classes in the way the model learned to do previously.
- **Step 3-Model Evaluation**: Now the model is ready to be assessed by its accuracy; how well fitted the data. If the measure is desirable, the process stops here. Otherwise, a new model is going to be selected (Step 1).

The family of models selected to solve the problem is the generalised linear models (GLMs) (Nelder & Wedderburn, 1972). GLMs are easily interpretable models because they describe a linear relationship between the response variable and the explanatory variables. However, there are some cases, such as binary classification, where there is no direct linear relationship between them. The literature on such a

kind problem refers that there is a linear relationship between the logistic form of odds ratio and the variables. Logistic regression model describes this relationship.

Let p(Y= "W" | X=x) and $[1 - \text{p(Y} = \text{"W"} \mid \text{X} = \text{x})]$ the probabilities of Win and Loss respectively, given the variable X= $X_1, \dots, X_p$ and the outcome Y=("W", "L"). The odds ratio is the probability of Win divided by the probability $1 - \text{p(Y} = \text{"W"} \mid \text{X} = \text{x})$of Loss. Therefore, the logistic model is described as follows:

$$\log\left(\frac{p(Y = \text{"W"} \mid X = x)}{1 - \text{p(Y} = \text{"W"} \mid \text{X} = \text{x})}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

where $\log(\frac{\text{p(Y= "W" | X=x)}}{\text{p(1−p(Y= "W" | X=x))}}$ = log (odds ratio) and the parameters of the variable X. These parameters are estimated using the Most Likelihood Estimation (MLE) method (Aldrich, 1997). To estimate the probability of Win, exponential is applied to the above equation and the equation becomes:

$$\text{p(Y} = \text{"W"} \mid \text{X} = \text{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}}$$

where $\text{p(Y} = \text{"W"} \mid \text{X} = \text{x})$ takes values from 0 to 1.

The observations get a predicted score in the range [0,1]. The score threshold to make the decision of classifying examples as 0 or 1 is set by default to be 0.5. This means that:

$$Fixture\ Result = \begin{cases} Win, & \text{p(Y} = \text{"W"} \mid \text{X} = \text{x}) > 0.5 \\ Loss, & \text{p(Y} = \text{"W"} \mid \text{X} = \text{x}) < 0.5 \end{cases}$$

A significant advantage of the Logistic Regression Model is that classifies not only the instances but also returns the probabilities that these instances belong to a class. This fact is advantageous, in particular when different tactics are designed during the pre-match approach and moreover when the online game stats may be used to make predictions during the match and adjust the tactics and the whole strategy.

When constructing a machine learning model, the objective is the model be as simple as it can be. Simplifying a model means that the model should be easily interpreted and use the least features need to explain the majority of the variance of the given dataset. This perspective is based on the Occam's Razor principle that "of two competing theories, the simpler explanation of an entity is to be preferred". William of Occam stated that "plurality should not be posited without necessity" (Duignan, 2015).

In cases where the feature set is vast, variable reduction while trying to maintain a high proportion of variance that the variables explain is substantial. Initially, the training set is used to train two models: the null model and the saturated model. The null model is the model that includes only the intercept and is interpreted as the model that supports the null hypothesis that no variable affects the game outcome. Instead, the saturated model describes the full set of variables that are used to explain the game outcome.

The objective is to find a model between these two models that better combines the performance of the full model (high proportion of explained variance) with the low cost of the null one (least variables). The method utilised for the model selection is the stepwise selection. During the execution of this method, variables are being iteratively added and deducted from a starting null model until there is no significant change in variance. It combines the properties of the forward selection, where the most contributed covariates are selected, and the backward selection (or better elimination) where after a variable induction round, the variables which are no longer provide an improvement to the model fit are eliminated. The quality indicator that is used in this study is the Akaike Information Criterion (AIC). AIC is a criterion which estimates the relative amount of information being lost due to the dimensionality of the models. A model achieves better accuracy when AIC value is low. Another similar to AIC criterion is the Bayesian Information Criterion (BIC). However, studies such as Burnham's & Anderson's (2004), Vrieze's (2012) and Yang's (2005) conclude that AIC has been advantageous over BIC in several trials.

Deployment/ Optimisation

Any successful Data Science project and its results should be able to be appropriately interpretable by the data scientists-storytellers and readily perceivable by the non-technical stakeholders. Even though the quality of the data analysis is very high, the objective is to steer the decision-maker into taking action (IMS Proschool, n.d.). It is essential that the analysis can answer the fundamental questions of the problem and to be reproducible so updates on the model may be applied. Therefore, clear and coherent communication of the results and the potentials of the analysis lead the decision-makers to trust the outcomes and engage themselves in deploying the model benefiting of its strengths.

No one can state that the perfect model exists. Instead, by applying the suggested model(s) to new unseen observations, the model will be continuously evaluated and updated to meet the expectations that the stakeholders have set.

Results

Data Mining

Before retrieving data from the SRU-created Power BI relational database, it was necessary to insert two new columns into the "All fixtures" table in order to record the game outcome for each team. These columns were generated by applying a conditional rule that compares the full-time score between the competitors of a match and ranks each team game outcome as Win (W), Draw (D), and Loss (L). After filtering the game outcome to isolate the draw results, the dataset in combination with the R-Script for the Model Training and Evaluation are extracted to R-Studio (see Appendix Part C, Script 1). The dataset consists of the competitions, the matchday, the rival teams and their scores along with the events captured during the games and the final outcome for both Home and Away Teams. The events are described by six categorical variables: Action Name, Action Type, Action Result, Qualifier 3, Qualifier 4 and Qualifier 5.

Data Pre-processing and Analysis

The SRU Performance Division provided the researcher with a database of 3,258 games across 25 competitions. As the access to the database was restricted -the database was saved and given into a Power BI file that only 150,000 rows can be obtained at most (Microsoft Corporation, 2019)- the choice of the games and the competitions was made in random. As summarised below, the retrieved games were 328 in total. Once the database contains a big set of games, one can assume that the population follows the normal distribution (central limit theorem). Thus, the random sample will be closely approximated by a normal distribution.

Table 2 provides with summarised info about the competitive teams and the actions that occurred during the games across four competitions: the Anglo-Welsh Cup, the Currie Cup and the Mitre 10 Cup Six Nations. Although it was not feasible to retrieve the whole population to examine if the sample follows the same distribution to the whole population, there are some indicators that support the previous statement. Applying basic statistics in the whole population into Power BI, some interesting facts are being extracted.

In particular, the average scores of the sample are approaching the average scores of the whole population. The correspondent averages for home teams and away teams derived from the sample are

31.47 and 25.32 points per game towards 27.84 and 21.62 points per game obtained from the whole population. Although the score averages are not identical, considering the range of scores in recorded games that is [0,94] and the points difference between Home Teams and Away Teams are almost the same in both the sample and the population (at about 6 points difference for Home Teams), one can make an assumption that the two comparable magnitudes present similarities on their distributions.

Column	Count
Fixture	328
Home Team	47
Away Team	46
Home Team Average Score	31.47
Away Team Average Score	25.32
Action Name	24
Action Result	92
Action Type	158
Qualifier 3	20
Qualifier 4	19
Qualifier 5	25
Home team Wins	209
Away Team Losses	119
Home Team Win/Loss Ratio	63.7%

Table 2 Sample Statistics

In order to perform the analysis, the Actions and Qualifiers (categorical variables) should be converted into numerical by being transformed into counts of combined occurrence. By a quick review, the potential combinations were approaching a number with rank 1013! This frame exceeds the standard capabilities of a modern computer, and of course, this creation is infeasible. So, initially, a proper technique was to explore the dataset for NA values. The results were that 17% of the dataset contains Further exploring each variable individually, Qualifier 3, Qualifier 4, Qualifier 5 include more than 60% NA values and therefore, they may not be considered are significant factors of a win/loss and these variables were eliminated (*Table 3*).

After creating the new dataset with the actual combinations of the events, the performance indicators were created (1085 variables). By applying the rule of 40 percent (the rule has been described in the

relative paragraph in the previous chapter), the dataset with the PIs were much more shortened, containing 210 features captured from 654 observations (323 matches; 2 teams per match).

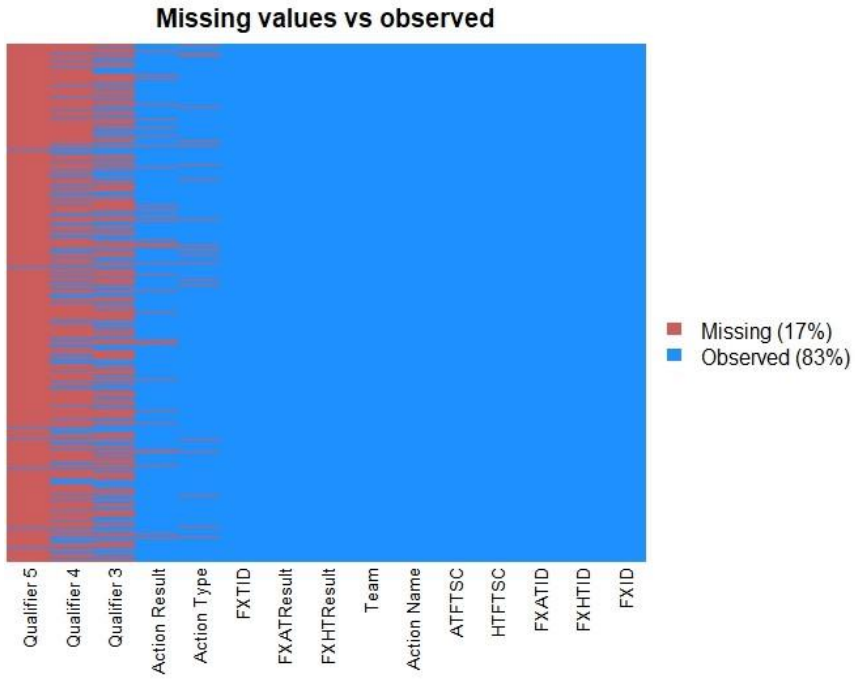

Figure 5 Observing missing values using missmap() (R-package: Amelia)

Attribute	Proportion of NA
Action Name	0%
Action Result	8.62%
Action Type	6.31%
Qualifier 3	62.38%
Qualifier 4	75.77%
Qualifier 5	98.61%

Table 3 Illustration of the NA values per attribute - Qualifiers 3,4 and 5 may not be considered as amongst the significant factors of a game result

After this step, the latter subject of data pre-processing was the treatment of the residual NA values in the dataset. As aforementioned in Methodology, the proper method for healing NA values was the substitution of NA values with the mode of each variable. At this point, the dataset could be deemed as clean and tidy.

Machine Learning Model Implementation

Two classifiers were created and compared at this stage. The first classifier (Classifier 1) was produced by the stepwise model selection and the second one (Classifier 2) was an enhanced version of Classifier 1 which was generated by combining the most statistically significant variables from Classifier 1 (Z-test; p-value <0.05) and the subset of them that reduced the deviance in the model (χ^2 test; p-value <0.05).

For the classifiers' validation, three sets were used:

- The training set: The training set is 75% of the final dataset and is used to train the classifiers.
- The test (evaluation) set: The test set is the remaining 25% of the final dataset and is used to examines the effectiveness of the classifiers.
- The prediction (validation) set: The prediction set is another obtained dataset from the database where selected games are games in which home teams have the highest Win/Loss ratio (see Appendix Part C, Script 2). Thus, that is going to be examined is the robustness of the classifiers when feeding with samples which are somehow biased.

The training set was used twice to fit data in two models: a) the null model, namely the model with only the interception considered for classifying the game outcome and b) the saturated model with all variables contributing to the game outcome. These two models were the inputs of the stepwise method. The output of this method was Classifier 1 (see Appendix Part A.1). Classifier 1 was trained and tested. This classifier was further simplified by the way described above, and so Classifier 2 was constructed (see Appendix Part A.2).

A comprehensive comparison of the candidate classifiers requires the assessment of the classifiers throughout the stages of training, test and predicting. In *Table 4* and *Table 5* the metrics for the two classifiers are illustrated.

	Classifier 1 (Training)	**Classifier 1 (Test)**	**Classifier 1 (Prediction)**
Accuracy	0.9265	0.7073	0.7422
95% CI	(0.8997, 0.948)	(0.6313, 0.7757)	(0.6989, 0.7821)
Cohen's Kappa	0.8531	0.4146	0.4843
Sensitivity	0.9306	0.7805	0.722
Specificity	0.9224	0.6341	0.7623
Prevalence	0.5	0.5	0.5
'Positive' Class	Win	Win	Win

Table 4 Classifier 1 metrics

The results of the assessment of Classifier 1 showed a contradicting image of the classifier's capabilities. The noticed Accuracy accompanied by the observed values of Cohen's Kappa and AUC

during the training process may be characterised as high and indicate a perfect classifier. However, the results of the test session did not confirm that this classifier can achieve the same level of successful classification. The Accuracy fell about 24%, the Accuracy Confidential Intervals (CI) were quite wider so that the variance was significantly higher, and Cohen's Kappa indicated that the classifier's ability is moderate. Observing the ROC graph (*Figure 6*), the curve was shortened indicating a classifier of medium accuracy.

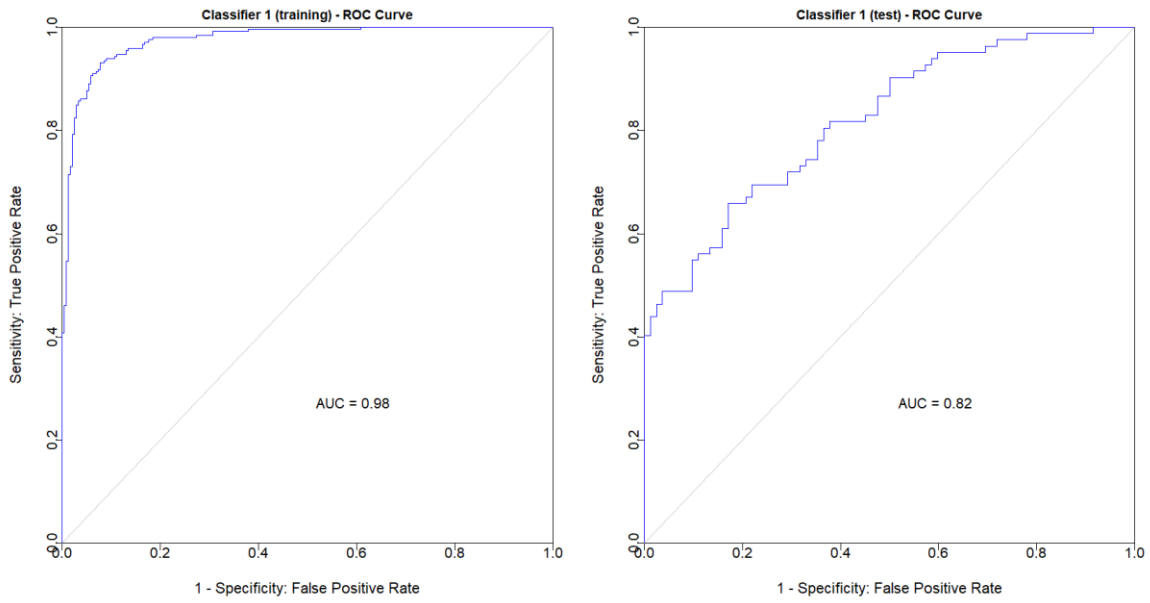

Figure 6 Receiver Operating Characteristic (ROC) graph and AUC value throughout the training and test sessions (Classifier 1)

As regards to Classifier 2, this classifier exposed high accuracy (>85%) and firm performance (Cohen's Kappa = 0.71) over the training period. The sensitivity and specificity measures are indicating high ability to categorise observations in both categories at the same level of Classifier 1 for the correspondent stage.

	Classifier 2 (Training)	**Classifier 2 (Test)**	**Classifier 2 (Prediction)**
Accuracy	0.8571	0.7317	0.778
95% CI	(0.823, 0.8869)	(0.657, 0.7978)	(0.7366, 0.8158)
Cohen's Kappa	0.7143	0.4634	0.5561
Sensitivity	0.8531	0.7927	0.7937
Specificity	0.8612	0.6707	0.7623
Prevalence	0.5	0.5	0.5
'Positive' Class	Win	Win	Win

Table 5 Classifier 2 metrics

However, similarly to Classifier 1, Classifier 2 did not confirm that it could predict the classes of the given unseen observations at the same level as during the training. A difference that was observed was

that although Classifier 2 had less preliminary accuracy than Classifier 2, however, all metrics values were slightly higher over the test period.

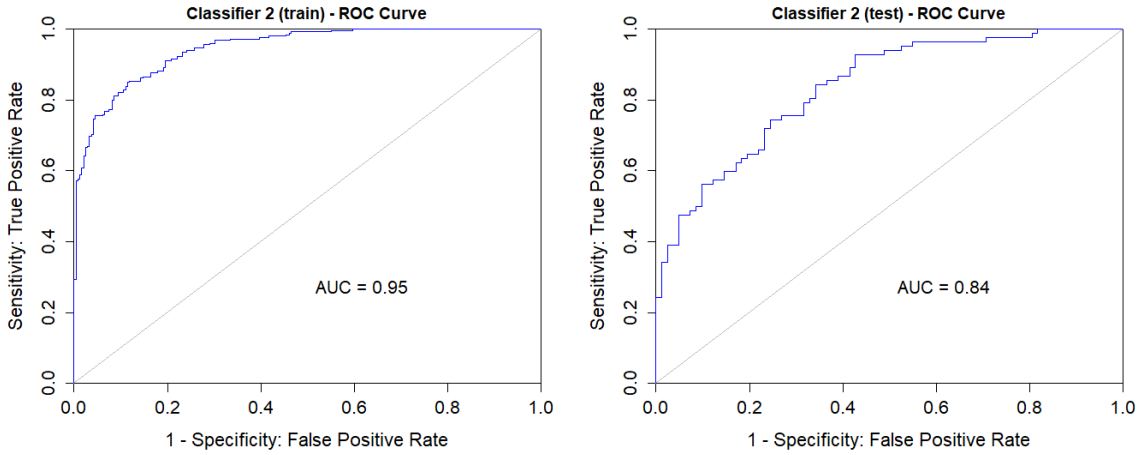

Figure 7 Receiver Operating Characteristic (ROC) graph and AUC value throughout the training and test sessions (Classifier 2)

In order to determine which of the two classifiers are better, the two classifiers were compared while trying to predict classes from a new dataset that has different characteristics from the already studied.

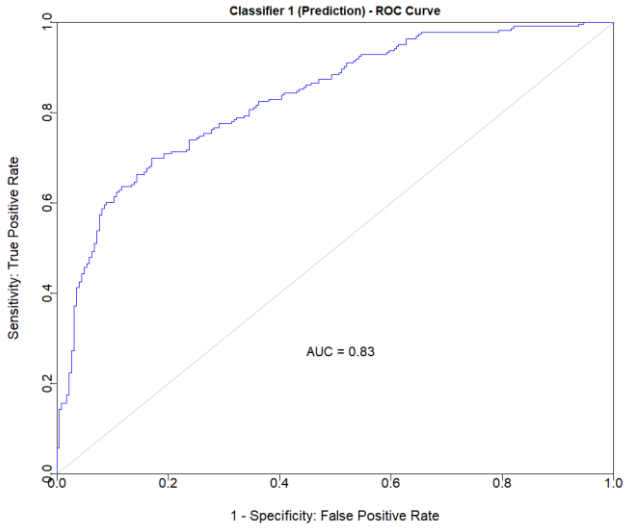

Figure 8 Receiver Operating Characteristic (ROC) graph and AUC value over the prediction session (Classifier 1)

Metrics from *Table 4* and *Table 5* supported by the ROC graphs (*Figure 8* and *Figure 9*) showed that these two classifiers had the same behaviour throughout the prediction session. Both increased their accuracy -compared to test predictions- and achieved better sensitivity and specificity. Classifier 2 has a better Cohen's κ value (0.56 towards 0.48), but it does make any substantial difference to the performance.

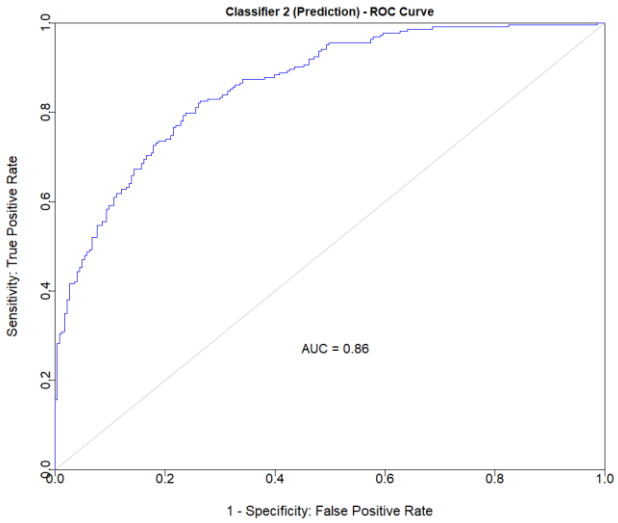

Figure 9 Receiver Operating Characteristic (ROC) graph and AUC value over the prediction session (Classifier 2)

Nevertheless, there is an excellent claim to select Classifier 2 instead of Classifier 1. This claim is that Classifier 2 leverages 45% fewer predictors (27 for Classifier 2 instead of 49 for Classifier 1), so that makes Classifier 2 a simpler model with similar or even better performance. The latter fact implies that Classifier 1 fails to generalise due to overfitting (too many predictors used for training) (Amazon Web Services, n.d.) and the deleted terms tend to explain more other individual variables included in the model rather than the game outcome (multicollinearity). Same issues raised for Classifier 2 as well but to a less extent.

Classifier 2 is composed of twenty-seven predictors explaining the variance of the sample. Twenty-two (out of twenty-seven) predictors are statistically significant to discriminate win results from losses. The twenty-seven predictors are composing the Key Performance Indicators grouped by levels of the first categorical variable used for the analysis, Action Name. So, the KPIs belong to the following game actions: 1) Sequence 2) Possession 3) Carry 4) Collection 5) Lineout 6) Kick 7) Penalty Conceded 8) Missed Tackle 9) Goal Kick and 10) Referee Review. These groups along with their correspondent features and their Odds Ratio (OR) are presented in *Table 6*.

As aforementioned in Methodology, OR for a feature is the ratio of the odds of a team to win the game given the feature and the odds of a team to lose the game given the same feature. So, when a feature has OR >1 means that if the feature's value increases one unit holding the others constant, the odds for winning due to this feature increases.

	Odds up		Odds down	
Action Name	Action_Name_Action_Type_Action_Result	OR	Action_Name_Action_Type_Action_Result	OR
Sequence			`Sequence_50m Restart`	0.407
			`Sequence_Tap Pen`	0.3005
Possession	`Possession_50m Restart`	1.9979	`Possession_Turnover Won_End Turnover`	0.2558
	`Possession_Turnover Won`	1.8615		
	`Possession_Lineout_End Try`	3.052		
	`Possession_Kick Return_End Set Kick In Play`	4.8129		
Carry	`Carry_Other Carry_Try Scored`	2.6399	`Carry_Other Carry_Error`	0.6712
	`Carry_Other Carry_Pass`	1.3948		
Collection	`Collection_Restart Catch`	3.5814	`Collection_Restart Catch_Success`	0.2674
Lineout			`Lineout Throw`	0.759
Kick	`Kick_Touch Kick_Kick In Touch (Full)`	2.8631		
	Kick_Low	1.4515		
	`Kick_Territorial_Kick In Touch (Full)`	3.9346		
Penalty Conceded	`Penalty Conceded_Scrum Offence_No Action`	13.1698	`Penalty Conceded_Scrum Offence`	0.0674
Missed Tackle	`Missed Tackle_Outpaced`	2.5179	`Missed Tackle`	0.537
Goal Kick	`Goal Kick`	1.4228		
Referee Review	`Ref Review`	1.3963		

Table 6 The most significant features of Classifier 2 (22 out of 27) and their odds ratio (OR). A feature reinforces the odds of winning when OR>1. Otherwise, the feature limits the odds of winning. The feature pattern Action_Name_Action_Type_Action_Result implies that Action_Name is the first component, Action_Type the second component and Action_Result the last component (if it exists).

For example, if a team carries the ball and scores a try, the log(odds) of this team winning the game increase by 164%. On the other hand, if OR<1 for a specific feature, then for each feature unit increment holding the others constant, the log(odds) for this team to win the match are limited. For example, if a penalty is conceded to a team due to a scrum offence, then the log(odds) for this team to win are reduced by 93.3%.

Summarising *Table 6*, twenty-two out of twenty-seven variables make a statically significant impact on the game outcome. In particular, possessions from 50m restart, possessions won from the opposition in a general way, tries started from gaining position after a lineout, kicks in play, tries and passes stemmed from other carry types, number of attempts to catch the ball from a restart kick, full kicks in touch, tackles that the attackers avoided, goal kicks and the intervention of referee during a match result in a 39-381% increase of the log(odds) for a winning result. Worth mentioning is that the number of conceded penalties due to a scrum offence without extra punishment can result in a 1,200% increase of log(odds) for a winning result. Conversely, starting a sequence from a 50m restart gained opposition or a tap penalty, losing the ball via an error when attacking, the number of carry errors, lineout throws, successful ball collections from a restart kick, conceded penalties due to a scrum offence and missed tackles may result in a 24-94.4% fall in the log-likelihood of winning a match.

Discussion

The identification of the KPIs which affect the game outcome provides the coaches with insight which may be leveraged to construct strategies and tactics that lead the teams and player to increase their performance either by developing new skills or training on specific parts of a play.

The study aimed to reduce the data set whilst retaining variance as possible, create an ML classifier that can adequately predict the outcome in Rugby Union games and identify those performance indicators that discriminate the game outcome. After retrieving data for 323 games plus a validation sample of the most successful home teams, two classifiers were constructed. After assessing these two classifiers, Classifier 2 selected. The selected classifier's Accuracy was 75.8% in average. Twenty-two performance indicators (out of twenty-seven that the model includes) compose ten groups associated with game actions. These performance indicators were divided into two categories according to whether to reinforce or limit the odds of winning.

The KPIs that increase the log(odds) of winning stemmed from actions that help teams maintain the possession, gain meters/territory and utilise individual player's skills (power, shot/pass accuracy, velocity) in order to score tries and goal kicks mainly. On the other hand, teams that fail to defend effectively or keep possession and make errors on the transition game from their side to the opponent's has fewer opportunities to win a match.

Previous studies seem to agree with the results as regards to the notions of possession ((Ortega, et al., 2009), (Vaz, et al., 2011), (Higham, et al., 2014) (Parmar, et al., 2018a)), and running in the open field ((Vaz, et al., 2011), (Woods, et al., 2017), (Parmar, et al., 2018a)) as elementary parts of success. Nevertheless, there are controversial findings on ways of scoring that matter and the lineouts as a factor of win or loss. This study discovered that goal kicks and tries coming from a carry are significant ways of scoring, and therefore winning, while Ortega and partners (2009) found that points scored by conversion and successful drops are most significant. As a goal kick is not always successful, this factor needs further analysis to investigate if only successful goal kicks matter or transformation to a goal kick success/miss ratio is a better manipulation of this indicator. Moreover, in this study, the number of lineouts is considered as a negative factor of win. This comes into contrast with the studies of Jones (2004) and (Woods, et al., 2017) in which the lineout success matters in the game outcome. Again, like the goal kick indicator, further analysis or transformation of the lineout feature is needed.

Some interesting comparisons and interpretations are extracted by this research. Sequence and possession are two similar terms that describe a period of time the ball is in play. The difference is that the sequence term does not investigate which team has the possession in contrast to possession that describes a period of time that a specific team holds the ball. In the results, the effect of these two terms is entirely different. Further analysis of these terms could explain which conditions brought this result.

Another contradictory finding that needs further analysis is the classification of successful ball collection after a restart kick. Successful ball collection is a way of gaining/retaining possession and likely gaining territory. So, this indicator may need a transformation to a success/failure ratio. According to the results, Referee Review has a positive impact on the probability of a team winning the game. However, it is not clear which types of actions associated with Referee Review are the reasons that this action forces the 'positive' effect on the game outcome. Finally, a penalty conceded due to a scrum infringement seems not to be clear on how it affects the game outcome. An interpretation of this result is that this infringement is used as a tactic from the teams in order to defend their side they prefer to take the risk of further punishment (with a yellow/red card, penalty kick, etc.).

If someone wants to compare the implemented classifier to approaches of previous studies, Classifier 2 metrics indicate that it is a competitive classifier in terms of accuracy. Classifier 2 achieved an accuracy of 75.8% in average, when other approaches achieve 75.5% (Repeated ANOVA measures (Vaz, et al., 2011)) and 90% (PCA, linear and logistic regression (Parmar, et al., 2018a)) respectively. However, these approaches have not measured by other binary classifiers' metrics, so sensibility, specificity or Cohen's Kappa are not known for these classifiers. Another point that is not known is what was the prevalence in the datasets used for training and evaluation. "Using data learning algorithms on skewed data sets can cause models to effectively predict correctly for the larger class while performing poorly for the minority class" (Morrison, 2016). That means that maybe the training dataset was prejudiced in favour, for example, of the winning outcome so that the model could be characterised of false sensitivity.

Conclusion

SRU Performance Division has set a problem: How can the performance analysts be informed of the most updated Key Performance Indicators through a sophisticated technology solution with accuracy and speed? After retrieving data for 323 games plus a validation sample of the most successful home teams, the author (considered as a data scientist intern) implemented a binary classifier that is based on a Logistic Regression model. The Logistic Regression model belongs to the family of generalised linear models and among its advantages are low complexity (thus, less computational resources), natural interpretation and the ability to manipulate any variables without being scaled.

The implemented classifier was optimised by using χ^2- and Z- statistical tests achieving a 75.8% accuracy. It compared to solutions in related problems from previous studies, and it was found that can stand up against other implementations. Twenty-two predictors are composing the Key Performance Indicators grouped by levels stemmed from the first categorical variable used for the analysis, Action Name. So, the KPIs belong to the following game actions: 1) Sequence 2) Possession 3) Carry 4) Collection 5) Lineout 6) Kick 7) Penalty Conceded 8) Missed Tackle 9) Goal Kick and 10) Referee Review.

These performance indicators were divided into two categories according to whether to reinforce or limit the odds of winning. In particular, possessions from 50m restart, possessions won from the opposition in a general way, tries started from gaining position after a lineout, kicks in play, tries and passes stemmed from other carry types, number of attempts to catch the ball from a restart kick, full kicks in touch, tackles that the attackers avoided, goal kicks and the intervention of the referee during a match result in a 39-381% increase of the log(odds) for a winning result. Worth mentioning is that the number of conceded penalties due to a scrum offence without extra punishment can result in a 1,200% increase of log(odds) for a winning result. Conversely, starting a sequence from a 50m restart gained opposition or a tap penalty, losing the ball via an error when attacking, the number of carry errors, lineout throws, successful ball collections from a restart kick, conceded penalties due to a scrum offence and missed tackles may result in a 24-94.4% fall in the log-likelihood of winning a match. Controversies occurred during the interpretation of the results may be treated either by variable transformation or further analysis of the variables.

Reflection Points / Further Improvement

Although the implemented classifier achieved high accuracy, there are reasons to believe that the logistic model or another Machine Learning implementation may bring better results if a sequence of issues is handled in the future. Changes should happen on input datasets and training session. Moreover, more engagement of the performance analysts could be beneficial for constructing a robust model.

- **Input Dataset**

  Managing NA values is a crucial factor for achieving effective variable selection. The rule of 40% applied during this project enabled the feature engineering, but results showed that some of the omitted variables might be used as predictors to interpret the results better. Other ways of preparing the data than transforming the features to numerical variables may be applicable.

- **Alternative training methods/model selection**
  - Reduce overfitting/ Change machine learning approach

    In this study, training and evaluation were conducted by randomly splitting the dataset into training and test sets. However, the classifier failed to manage the phenomenon of overfitting. A suggested way to improve accuracy and reduce the chance of overfitting is the k-fold cross-validation technique ((Wikipedia contributors, n.d.)). Cross-validation technique is a method that creates several training and test sets by just initially dividing the dataset into k segment. Each segment takes the role of the test set while the others are used as the training set. This is an iterative process, and each instance belongs to one segment. In each rotation, the model trained by the k-1 segments is evaluated using the remaining dataset. Another way is to increase the training epochs (iterations) in order to achieve higher convergence of Maximum Likelihood Estimation which is the metric used for model training. Other interventions are batch data processing where data is being sliced into batches and feed the model sequentially (Walker, 2013) and construction of other ML models such as Random Forests, SVM and Neural Networks.
  - Reduce collinearity

    Collinearity happens when individual variables have strong correlations and better explain each other than the game outcome. The treatment of this effect is the application of sophisticated dimension reduction techniques. The applied statistical tests tried to manage this problem, but currently, machine learning offers very efficient methods for dimension reduction, such as Linear Discriminant Analysis (LDA), Principal Components Analysis (PCA), k-means and K-NN. These methods not only reduce the dimensions of the features but also create clusters and classes so that KPIs can be explained by specific PIs.

- **SRU performance analysts' intervention**

  SRU Performance Division is the owner of the dataset. So, SRU performance analysts are prompted to check the database for faults and communicate with OPTA to fix them. Some teams seem not to be registered with an ID on table ***All Clubs***, e.g. Italy A. Also, Teams from table ***All Matchdata*** are not associated with Fixtures and Results. For this reason, the

researcher performed the analysis using a sample based on Home Teams. Once this issue gets resolved, teams which either play home or away can be analysed individually.

The development of a model that describes KPIs affecting the game outcome at Rugby Union fixtures is now feasible. Contemporary computer systems and sophisticated methods can be leveraged in order to Scottish Rugby Union take advantage over its competitors and consult the teams of its membership, so they further improve the quality of their players and consequently the game quality, and the Rugby continues to thrive and grow in Scotland.

Appendix

Abbreviations:

All Fixtures: Sport Events on particular dates (Table)

FXID: ID of the Fixture (Competition)

FXHTID: Home Team ID of the Fixture

FXATID: Away Team ID of the Fixture

HTFTSC: Home Team Full Time Score

ATFTSC: Away Team Full Time Score

FXHTResult: Home Team Fixture Result (Win: "W", Loss: "L")

FXATResult: Away Team Fixture Result (Win: "W", Loss: "L")

FXResult: Team Fixture Result (Win: "W", Loss: "L")

All Matchdata: Data from all matches (Table)

FXID: Foreign key from ***All Fixtures***

ID: ID from facts occurred throughout a Fixture

All names used to describe Action Names, Action Types and Action Results had been given by OPTA to SRU Performance Division and then were confidentially available to the researcher.

Part A – Summary of classifiers

A.1 Classifier 1

Formula:

```
FXResult ~ `Sequence_50m Restart` + `Possession_50m Restart` +
    `Carry_Other Carry_Try Scored` + `Missed Tackle` +
    `Sequence_Tap Pen` + `Possession_Turnover Won` +
    `Carry_Other Carry_Error` + `Collection_Restart Catch` +
```

```
    `Possession_Lineout_End Try` + `Lineout Throw` +
    `Kick_Touch Kick_Kick In Touch (Full)` + `Penalty Conceded_Scrum Offen
ce` +
    `Penalty Conceded_Scrum Offence_No Action` + `Collection_Restart Catch
_Success` +
    `Ref Review` + Kick_Low + `Goal Kick` + `Possession_Kick Return_End Se
t Kick In Play` +
    `Possession_Turnover Won_End Turnover` + `Carry_Other Carry_Pass` +
    `Missed Tackle_Outpaced` + `Penalty Conceded_Not Releasing_No Action`
+
    `Kick_Territorial_Kick In Touch (Full)` + `Tackle_Line Tackle_Turnover
Won` +
    `Possession_Turnover Won_End Set Kick In Play` + Try +
    `Tackle_Guard Tackle` + `Collection_Defensive Loose Ball_Success` +
    Ruck + `Tackle_Line Tackle_Sack` + `Lineout Throw_Throw Back` +
    `Lineout Take_Lineout Win Back_Won Clean` + `Attacking Qualities` +
    `Lineout Throw_Throw Middle` + `Attacking Qualities_Initial Break` +
    `Lineout Take_Lineout Win Front_Won Clean` + Pass_Offload +
    Pass + `Defensive Scrum` + `Defensive Scrum_Scrum Half Pass` +
    `Missed Tackle_Bumped Off` + Turnover + `Possession_Start Set Lineout
Steal` +
    `Carry_Other Carry_Kick` + `Carry_Pick And Go_Off Load` +
    `Kick_Territorial_Collected Bounce` + Kick_Chip + `Kick_Territorial_Ca
ught Full` +
    Scrum + Possession_Scrum - 1


Coefficients:                          Estimate Std. Error z value Pr(>|z|)


 `Sequence_50m Restart`                -1.2276     0.2055  -5.973 2.33e-09 ***
 `Possession_50m Restart`               1.1017     0.2168   5.080 3.76e-07 ***
 `Carry_Other Carry_Try Scored`         1.3237     0.3878   3.413 0.000642 ***
 `Missed Tackle`                       -0.5693     0.1971  -2.889 0.003863 **
 `Sequence_Tap Pen`                    -1.5767     0.2959  -5.328 9.94e-08 ***
 `Possession_Turnover Won`              1.0745     0.1868   5.751 8.86e-09 ***
 `Carry_Other Carry_Error`             -0.3307     0.1784  -1.854 0.063750 .
 `Collection_Restart Catch`             2.1263     0.4544   4.679 2.88e-06 ***
 `Possession_Lineout_End Try`           1.4658     0.5226   2.805 0.005032 **
 `Lineout Throw`                       -0.8261     0.1677  -4.927 8.35e-07 ***
 `Kick_Touch Kick_Kick In Touch (Full)`  1.7396     0.4711   3.693 0.000222 ***
 `Penalty Conceded_Scrum Offence`      -5.7166     1.2186  -4.691 2.72e-06 ***
 `Penalty Conceded_Scrum Offence_No Action`  5.1702     1.2119   4.266 1.99e-05 ***
 `Collection_Restart Catch_Success`    -1.5787     0.7666  -2.059 0.039458 *
 `Ref Review`                           0.6915     0.1910   3.621 0.000293 ***
 Kick_Low                               0.7178     0.2034   3.529 0.000417 ***
 `Goal Kick`                            0.5761     0.2338   2.464 0.013749 *
 `Possession_Kick Return_End Set Kick In Play`  2.3752     0.7483   3.174 0.001503 **
 `Possession_Turnover Won_End Turnover`  -2.1439     0.6855  -3.128 0.001762 **
 `Carry_Other Carry_Pass`               0.3812     0.1860   2.050 0.040384 *
 `Missed Tackle_Outpaced`               0.9719     0.4755   2.044 0.040956 *
 `Penalty Conceded_Not Releasing_No Action`  -7.0806     2.3209  -3.051 0.002282 **
 `Kick_Territorial_Kick In Touch (Full)`  3.6069     0.9654   3.736 0.000187 ***
 `Tackle_Line Tackle_Turnover Won`      1.3658     0.3348   4.080 4.50e-05 ***
 `Possession_Turnover Won_End Set Kick In Play`  -2.5112     1.0007  -2.510 0.012090 *
 Try                                    2.2495     1.2310   1.827 0.067630 .
```

```
`Tackle_Guard Tackle`                          -0.2183     0.1046  -2.088 0.036808 *
`Collection_Defensive Loose Ball_Success`       4.6173     1.6802   2.748 0.005993 **
Ruck                                           -0.7478     0.2524  -2.962 0.003052 **
`Tackle_Line Tackle_Sack`                       0.9046     0.4873   1.857 0.063378 .
`Lineout Throw_Throw Back`                      1.1602     0.2813   4.125 3.72e-05 ***
`Lineout Take_Lineout Win Back_Won Clean`       -1.3467    0.4209  -3.199 0.001377 **
`Attacking Qualities`                          -0.7857     0.3028  -2.595 0.009461 **
`Lineout Throw_Throw Middle`                    0.7489     0.2221   3.373 0.000744 ***
`Attacking Qualities_Initial Break`            -2.5321     1.6177  -1.565 0.117518
`Lineout Take_Lineout Win Front_Won Clean`      0.6022     0.2875   2.095 0.036206 *
Pass_Offload                                   -1.0675     0.3433  -3.110 0.001873 **
Pass                                            0.4788     0.1411   3.392 0.000693 ***
`Defensive Scrum`                               0.3983     0.1401   2.842 0.004480 **
`Defensive Scrum_Scrum Half Pass`              -2.4549     1.1960  -2.053 0.040106 *
`Missed Tackle_Bumped Off`                     -1.1631     0.4135  -2.813 0.004915 **
Turnover                                       -0.3562     0.1504  -2.368 0.017879 *
`Possession_Start Set Lineout Steal`            0.8422     0.2812   2.995 0.002742 **
`Carry_Other Carry_Kick`                        1.0118     0.4863   2.080 0.037485 *
`Carry_Pick And Go_Off Load`                   -0.7381     0.3496  -2.111 0.034730 *
`Kick_Territorial_Collected Bounce`            -0.9489     0.4693  -2.022 0.043200 *
Kick_Chip                                      -0.3634     0.2455  -1.480 0.138808
`Kick_Territorial_Caught Full`                  0.9536     0.4655   2.049 0.040505 *
Scrum                                           0.2303     0.1047   2.200 0.027819 *
Possession_Scrum                               -0.2420     0.1379  -1.756 0.079173 .

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 679.28  on 490  degrees of freedom
Residual deviance: 206.78  on 440  degrees of freedom
AIC: 306.78

Number of Fisher Scoring iterations: 8
```

Covariates	OR	Lower 95% CI	Upper 95% CI
`Sequence_50m Restart`	0.293	0.1959	0.4383
`Possession_50m Restart`	3.0093	1.9673	4.6031
`Carry_Other Carry_Try Scored`	3.7574	1.757	8.0355
`Missed Tackle`	0.5659	0.3846	0.8327
`Sequence_Tap Pen`	0.2067	0.1157	0.3692
`Possession_Turnover Won`	2.9286	2.0306	4.2237
`Carry_Other Carry_Error`	0.7184	0.5065	1.0191
`Collection_Restart Catch`	8.3838	3.4404	20.43
`Possession_Lineout_End Try`	4.331	1.5551	12.0617
`Lineout Throw`	0.4378	0.3152	0.6081
`Kick_Touch Kick_Kick In Touch (Full)`	5.6948	2.2621	14.3366
`Penalty Conceded_Scrum Offence`	0.0033	3.00E-04	0.036

`Penalty Conceded_Scrum Offence_No Action`	175.9576	16.3623	1892.225
`Collection_Restart Catch_Success`	0.2062	0.0459	0.9264
`Ref Review`	1.9968	1.3734	2.9032
Kick_Low	2.05	1.3759	3.0543
`Goal Kick`	1.7792	1.125	2.8137
`Possession_Kick Return_End Set Kick In Play`	10.7535	2.4807	46.6157
`Possession_Turnover Won_End Turnover`	0.1172	0.0306	0.4492
`Carry_Other Carry_Pass`	1.464	1.0168	2.1079
`Missed Tackle_Outpaced`	2.6429	1.0407	6.7115
`Penalty Conceded_Not Releasing_No Action`	8.00E-04	0	0.0756
`Kick_Territorial_Kick In Touch (Full)`	36.8505	5.5551	244.4507
`Tackle_Line Tackle_Turnover Won`	3.9188	2.0333	7.5526
`Possession_Turnover Won_End Set Kick In Play`	0.0812	0.0114	0.5772
Try	9.4834	0.8495	105.8725
`Tackle_Guard Tackle`	0.8039	0.6549	0.9867
`Collection_Defensive Loose Ball_Success`	101.2222	3.7592	2725.54
Ruck	0.4734	0.2887	0.7764
`Tackle_Line Tackle_Sack`	2.471	0.9508	6.4215
`Lineout Throw_Throw Back`	3.1907	1.8384	5.5378
`Lineout Take_Lineout Win Back_Won Clean`	0.2601	0.114	0.5935
`Attacking Qualities`	0.4558	0.2518	0.8251
`Lineout Throw_Throw Middle`	2.1147	1.3685	3.2679
`Attacking Qualities_Initial Break`	0.0795	0.0033	1.894
`Lineout Take_Lineout Win Front_Won Clean`	1.8262	1.0395	3.2084
Pass_Offload	0.3439	0.1755	0.674
Pass	1.6141	1.224	2.1285
`Defensive Scrum`	1.4893	1.1316	1.96
`Defensive Scrum_Scrum Half Pass`	0.0859	0.0082	0.8954
`Missed Tackle_Bumped Off`	0.3125	0.1389	0.7029
Turnover	0.7003	0.5215	0.9404
`Possession_Start Set Lineout Steal`	2.3214	1.3379	4.028
`Carry_Other Carry_Kick`	2.7505	1.0603	7.135
`Carry_Pick And Go_Off Load`	0.478	0.2409	0.9484
`Kick_Territorial_Collected Bounce`	0.3872	0.1543	0.9715
Kick_Chip	0.6953	0.4298	1.1249
`Kick_Territorial_Caught Full`	2.5949	1.0421	6.4617
Scrum	1.259	1.0254	1.5458
Possession_Scrum	0.785	0.5991	1.0286

Table 7 Table of Odds ratio for each covariate (Classifier 1)

A.2 Classifier 2

Formula:

```
FXResult ~ `Sequence_50m Restart` + `Possession_50m Restart` +
    `Carry_Other Carry_Try Scored` + `Missed Tackle` + `Sequence_Tap Pen`
+
```

```
    `Possession_Turnover Won` + `Carry_Other Carry_Error` + `Collection_Re
start Catch` +
    `Possession_Lineout_End Try` + `Lineout Throw` + `Kick_Touch Kick_Kick
In Touch (Full)` +
    `Penalty Conceded_Scrum Offence` + `Penalty Conceded_Scrum Offence_No
Action` +
    `Collection_Restart Catch_Success` + `Ref Review` + Kick_Low +
    `Goal Kick` + `Possession_Kick Return_End Set Kick In Play` +
    `Possession_Turnover Won_End Turnover` + `Carry_Other Carry_Pass` +
    `Missed Tackle_Outpaced` + `Penalty Conceded_Not Releasing_No Action`
+
    `Kick_Territorial_Kick In Touch (Full)` + Ruck + `Lineout Take_Lineout
Win Back_Won Clean` +
    `Lineout Throw_Throw Middle` + `Defensive Scrum_Scrum Half Pass`


Coefficients:                       Estimate Std. Error z value Pr(>|z|)


 (Intercept)                          16.96677  732.05995   0.023 0.981509
 `Sequence_50m Restart`               -0.89891    0.14609  -6.153 7.60e-10 ***
 `Possession_50m Restart`              0.69208    0.14189   4.877 1.07e-06 ***
 `Carry_Other Carry_Try Scored`        0.97074    0.30485   3.184 0.001451 **
 `Missed Tackle`                      -0.62174    0.12393  -5.017 5.25e-07 ***
 `Sequence_Tap Pen`                   -1.20225    0.21487  -5.595 2.20e-08 ***
 `Possession_Turnover Won`             0.62139    0.11444   5.430 5.65e-08 ***
 `Carry_Other Carry_Error`            -0.39864    0.13926  -2.862 0.004203 **
 `Collection_Restart Catch`            1.27576    0.29845   4.275 1.91e-05 ***
 `Possession_Lineout_End Try`                1.11580    0.41071   2.717 0.006593 **
 `Lineout Throw`                            -0.27580    0.08936  -3.087 0.002025 **
 `Kick_Touch Kick_Kick In Touch (Full)`      1.05189    0.33884   3.104 0.001907 **
 `Penalty Conceded_Scrum Offence`           -2.69762    0.81597  -3.306 0.000946 ***
 `Penalty Conceded_Scrum Offence_No Action`  2.57792    0.84080   3.066 0.002169 **
 `Collection_Restart Catch_Success`         -1.31913    0.58686  -2.248 0.024589 *
 `Ref Review`                                0.33379    0.13223   2.524 0.011592 *
 Kick_Low                                    0.37258    0.15115   2.465 0.013702 *
 `Goal Kick`                                 0.35259    0.17534   2.011 0.044339 *
 `Possession_Kick Return_End Set Kick In Play`  1.57130    0.54925   2.861 0.004225 **
 `Possession_Turnover Won_End Turnover`     -1.36325    0.48733  -2.797 0.005152 **
 `Carry_Other Carry_Pass`                    0.33274    0.14440   2.304 0.021204 *
 `Missed Tackle_Outpaced`                    0.92343    0.35629   2.592 0.009547 **
 `Penalty Conceded_Not Releasing_No Action`  -17.44747  732.05525  -0.024 0.980985
 `Kick_Territorial_Kick In Touch (Full)`     1.36982    0.60944   2.248 0.024596 *
 Ruck                                       -0.27948    0.19487  -1.434 0.151527
 `Lineout Take_Lineout Win Back_Won Clean`   -0.28965    0.25601  -1.131 0.257897
 `Lineout Throw_Throw Middle`                0.07003    0.15036   0.466 0.641420
 `Defensive Scrum_Scrum Half Pass`          -0.92754    0.78307  -1.184 0.236219


Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 679.28  on 489  degrees of freedom
Residual deviance: 289.44  on 462  degrees of freedom
AIC: 345.44
```

```
Number of Fisher Scoring iterations: 15
```

Covariates	OR	Lower 95% CI	Upper 95% CI
(Intercept)	23365562.51	0	Inf
`Sequence_50m Restart`	0.407	0.3057	0.5419
`Possession_50m Restart`	1.9979	1.5128	2.6385
`Carry_Other Carry_Try Scored`	2.6399	1.4524	4.7982
`Missed Tackle`	0.537	0.4212	0.6846
`Sequence_Tap Pen`	0.3005	0.1972	0.4579
`Possession_Turnover Won`	1.8615	1.4875	2.3296
`Carry_Other Carry_Error`	0.6712	0.5109	0.8819
`Collection_Restart Catch`	3.5814	1.9953	6.4283
`Possession_Lineout_End Try`	3.052	1.3645	6.8264
`Lineout Throw`	0.759	0.6371	0.9043
`Kick_Touch Kick_Kick In Touch (Full)`	2.8631	1.4737	5.5624
`Penalty Conceded_Scrum Offence`	0.0674	0.0136	0.3336
`Penalty Conceded_Scrum Offence_No Action`	13.1698	2.5344	68.4362
`Collection_Restart Catch_Success`	0.2674	0.0846	0.8447
`Ref Review`	1.3963	1.0775	1.8094
Kick_Low	1.4515	1.0793	1.952
`Goal Kick`	1.4228	1.009	2.0063
`Possession_Kick Return_End Set Kick In Play`	4.8129	1.6401	14.1233
`Possession_Turnover Won_End Turnover`	0.2558	0.0984	0.6648
`Carry_Other Carry_Pass`	1.3948	1.051	1.8511
`Missed Tackle_Outpaced`	2.5179	1.2525	5.0619
`Penalty Conceded_Not Releasing_No Action`	0	0	NA
`Kick_Territorial_Kick In Touch (Full)`	3.9346	1.1916	12.9916
Ruck	0.7562	0.5161	1.1079
`Lineout Take_Lineout Win Back_Won Clean`	0.7485	0.4532	1.2363
`Lineout Throw_Throw Middle`	1.0725	0.7987	1.4401
`Defensive Scrum_Scrum Half Pass`	0.3955	0.0852	1.8353

Table 8 Table of Odds ratio for each covariate (Classifier 2)

Part B – Confusion Matrices

Training Set

Confusion Matrix Classifier 1			
		Reference	
		Loss	Win
Prediction	Loss	226	17
	Win	19	228

Confusion Matrix Classifier 2			
		Reference	
		Loss	Win
Prediction	Loss	211	36
	Win	34	209

Test Set

Classifier 1			
		Reference	
		Loss	Win
Prediction	Loss	52	30
	Win	18	64

Confusion Matrix Classifier 2			
		Reference	
		Loss	Win
Prediction	Loss	55	27
	Win	17	65

Prediction Set

Confusion Matrix Classifier 1			
		Reference	
		Loss	Win
Prediction	Loss	211	36
	Win	34	209

Confusion Matrix Classifier 2			
		Reference	
		Loss	Win
Prediction	Loss	170	53
	Win	46	177

Part C – R Scripts

Script 1 – Model Training version 1.6.R

```
#Model Training version 1.6
#Author: Sokratis Dimitrios Chronopoulos
#Student ID: 201755818
# Input load. Please do not change # This dataset is automatically created by Power BI
`dataset` = read.csv('C:/Users/soc_x/REditorWrapper_c5186d63-559c-4985-82b4-8654ef12956e/input_df_e5a0227b-
9eda-486d-a3ed-55063851ab6a.csv', check.names = FALSE, encoding = "UTF-8", blank.lines.skip = FALSE);
#Check if used packages have been installed in this R session
list.of.packages <- c("dplyr", "plyr", "reshape", "Amelia", "caTools", "e1071", "caret", "tidyverse")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages, force=FALSE)
library(dplyr)
library(plyr)
library(Amelia) # for missing values map
library(reshape)
library(caTools) # for sample.split
library(e1071)
library(caret)
library(tidyverse)
library(broom)
# Function for replacing NAs with modes
Mode <- function (x) {
xtab <- table(x)
xmode <- as.numeric(names(which(xtab == max(xtab))))
if (length(xmode) > 1) xmode <- ceiling(mean(xmode)) #if there are more than one mode, then return the
average
return(xmode)
}
#Setting the current directory as working directory
path <- rstudioapi::getActiveDocumentContext()$path
setwd(dirname(path))
###### Exploratory Data Analysis #######
#Produce a map with missing values vs observed in raw dataset
missmap(dataset, main = "Missing values vs observed")
#calculate NA values proportion per variable from dataframe (raw data): True=NAs, False=Non-NAs
for (i in 1:(length(dataset[-(1:5)]))){
temp <- plyr::count(is.na(dataset[i+5]))
temp$freq<-(temp$freq/nrow(dataset))*100
print(temp)
}
#Create a new dataframe composed of the events occured for each team per fixture
attach(dataset)
unique_features <- as.data.frame(unique(data.frame(FXID,Team,FXHTID,FXATID,HTFTSC,ATFTSC,FXHTResult,
FXATResult)))
detach(dataset)
#Create and populate a new column called FXResult
unique_features$FXResult<-factor(NA,levels = c("L","W"), ordered = FALSE)
for (fixture in unique_features$FXID){
for (team in unique_features$Team[unique_features$FXID==fixture]){
if (Reduce("&",lapply(strsplit(team," ")[[1]],grepl,unique_features$FXHTID[unique_features$FXID==fixture &
unique_features$Team==team]))){
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]<-
unique_features$FXHTResult[unique_features$FXID==fixture & unique_features$Team==team]
}
else if (Reduce("&",lapply(strsplit(team,"
")[[1]],grepl,unique_features$FXATID[unique_features$FXID==fixture & unique_features$Team==team]))){
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]<-
unique_features$FXATResult[unique_features$FXID==fixture & unique_features$Team==team]
}
}
}
#Print the proportions of W/L in terms of aggregation and Home and Away Teams Performance Ratio on the
given dataset
" for testing purpose / future use : inserts FXResult into original dataset
dataset$FXResult<-factor(NA,levels = c('L','W'), ordered = FALSE)
for (fixture in unique(dataset$FXID)){
for (team in unique(dataset$Team[dataset$FXID==fixture])){
dataset$FXResult[dataset$FXID==fixture&dataset$Team==team]<-
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]
}}"
"Create 6 levels with counts of different combinations occuring from the following variables from each team
of the occured fixtures:
Level 1: count(Action Name, Action Type, Action Result, Qualifier 3, Qualifier 4, Qualifier 5)
Level 2: count(Action Name, Action Type, Action Result, Qualifier 3, Qualifier 4)
Level 3: count(Action Name, Action Type, Action Result, Qualifier 3)
Level 4: count(Action Name, Action Type, Action Result)
Level 5: count(Action Name, Action Type)
Level 6: count(Action Name)
"
```

```
#level1<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`', '`Qualifier 4`', '`Qualifier 5`'))
#level2<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`', '`Qualifier 4`'))
#level3<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`'))
level4<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`'))
level5<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`'))
level6<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`'))
#Now we are going to transform these matrices and merge the features to receive unique values of these
combinations
#level1_final <- cast(level1,
FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3+Qualifier.4+Qualifier.5)
#level2_final <- cast(level2,
FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3+Qualifier.4)
#level3_final <- cast(level3, FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3)
level4_final <- cast(level4, FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result)
level5_final <- cast(level5, FXID+FXHTID+Team~Action.Name+Action.Type)
level6_final <- cast(level6, FXID+FXHTID+Team~Action.Name)
#levels_list <- list(level1_final,level2_final,level3_final,level4_final,level5_final,level6_final)
levels_list <- list(level4_final,level5_final,level6_final)
#Now, we can safely remove the individual dataframes we have created in prior to finalise the dataframe
will be used for producing the correlations
rm(level1,level2,level3,level4,level5,level6)
rm(level1_final,level2_final,level3_final,level4_final,level5_final,level6_final)
#Here, we create the final dataframe named global_table that includes all the instances occurring for each
team during the selected fixtures
global_table <- levels_list[[1]] #Fill in the table with the first level according to the order that levels
have been inserted into teh list
for (i in 2:length(levels_list)) # Continue the fullfilment using a for-loop for the remaining levels
{
global_table <- merge(global_table,levels_list[[i]],by= c('FXID','FXHTID','Team'))
}
#calculate NA values proportion per variable level from dataframe (global_table): True=NAs, False=Non-NAs
for (i in 1:(length(global_table[-(1:3)]))){
temp <- plyr::count(is.na(global_table[i+3]))
temp$freq<-(temp$freq/nrow(global_table))*100
print(temp)
}
#Create the dataframe with the tranformed data used to make the Machine Learning Model
global_table$FXResult<-factor(NA,levels = c("L","W"), ordered = FALSE)
for (fixture in global_table$FXID){
for (team in global_table$Team[global_table$FXID==fixture]){
if (Reduce("&",lapply(strsplit(team," ")[[1]],grepl,unique_features$FXHTID[unique_features$FXID==fixture &
unique_features$Team==team]))){
global_table$FXResult[global_table$FXID==fixture & global_table$Team==team]<-
as.factor(unique_features$FXHTResult[unique_features$FXID==fixture])
}
else if (Reduce("&",lapply(strsplit(team,"
")[[1]],grepl,unique_features$FXATID[unique_features$FXID==fixture & unique_features$Team==team]))){
global_table$FXResult[global_table$FXID==fixture & global_table$Team==team]<-
as.factor(unique_features$FXATResult[unique_features$FXID==fixture])
}
}
}
write.csv(global_table,"Global Table.csv",row.names=FALSE) #Export the dataframe to a .csv file
#global_table <- read.csv("Global Table.csv")
#Create a reduced global table removing all columns and rows including NAs in more than 40%
g_other<- global_table[, colSums(is.na(global_table)) < 0.4*(nrow(global_table)-1)] #columns
g_other<-g_other[!(rowSums(is.na(g_other))/ncol(g_other)>0.4),]
#Remove redudant columns
#(The 3-variable columns which end with 'NA' are equal to the 2-variable columns
#that share the first two variables)
g_other<-select(g_other, -ends_with("NA"))
#Remove non-numeric columns (e.g., FXID)
g_other<-g_other[-(1:3)]
# Cleaning NAs using central tendency: Mode
for (var in 1:ncol(g_other)) {
g_other[is.na(g_other[,var]),var] <- Mode(g_other[,var])
}
###### Machine Learning Implementation #######
#Split data into training and test sets
set.seed(101) #for reproducibility
sample = sample.split(g_other$FXResult, SplitRatio = .75) #splitting data..
train = subset(g_other, sample == TRUE) #Training set: 75%
test = subset(g_other, sample == FALSE) #Test set: 25%
#Fit a logistic model with no variables accounted for this model: Null model
model_0<- glm(FXResult~ 1, data=train, family="binomial")
#Fit a logistic model with all data from all variables: Full model
model_full <- glm(FXResult~. , data=train, family="binomial")
#Find the best model using stepwise selection
stepwise = step(model_0,
scope=list(lower=formula(model_0),upper=formula(model_full)), direction="both")
#Remove the intercept from the model -> there is no chance to Win if all variables are equal to 0
stepwise <- update(stepwise, .~. -1)
#save model parameters in a .csv file
tidy_stepwise<- tidy(stepwise)
```

```
write.csv(tidy_stepwise,"Model 1.csv",row.names=FALSE)
#Save odds ratios into a file
odd_ratios_stepwise<-data.frame(c(round(exp(stepwise$coefficients),4)))
lower.ci <- round(odd_ratios_stepwise*exp(-1.96*tidy_stepwise$std.error),4)
upper.ci<- round(odd_ratios_stepwise*exp(1.96*tidy_stepwise$std.error),4)
odd_ratios_stepwise <-cbind(odd_ratios_stepwise,lower.ci, upper.ci)
colnames(odd_ratios_stepwise) <- c("OR", "Lower 95% CI", "Upper 95% CI")
write.csv(odd_ratios_stepwise,"Odds ratios (Classifier 1).csv",row.names=TRUE)
#Save the model for future use (apply to a dataset of unseen observations)
saveRDS(stepwise, file = "lg1.RDS")
# Evaluate the model using the training set
fitted_probs <- predict(stepwise,newdata=train,type='response')
fitted_results <- ifelse(fitted_probs > 0.5,"W","L")
fitted_results <- factor(fitted_results,levels = c("L","W"), ordered = FALSE)
misClasificError_train <- mean(fitted_results != train$FXResult)
confusionMatrix(fitted_results, train$FXResult, positive="W")
# Creating performance object (sources:http://scaryscientist.blogspot.com/2016/03/roc-receiver-operating-
characteristics.html, https://github.com/chupvl/R_scripts/blob/master/rocr_wLegend.R accessed: 14/03/2019)
library("ROCR")
#setting visual parameters
par(mar=c(5,5,2,2),xaxs = "i",yaxs = "i",cex.axis=1.3,cex.lab=1.4)
par(mfrow=c(1,2)) #Illustrate 2 plots side by side
# calculating the values for ROC curve
perf.obj_train <- prediction(predictions=fitted_probs, labels=train$FXResult)
# Get data for ROC curve
roc.obj_train <- performance(perf.obj_train, measure="tpr", x.measure="fpr")
plot(roc.obj_train,
main="Classifier 1 (training) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc <- performance(perf.obj_train, measure = "auc")
auc <- auc@y.values[[1]]
# adding ROC AUC to the center of the plot
maxauc<-max(round(auc, digits = 2))
maxauct <- paste(c("AUC = "),maxauc,sep="")
legend(0.37,0.34, c(maxauct),border="white",cex=1.33,box.col = "white")
#Evaluate the model using the test set
#Predict likelihood of outcome = "W" and convert them to the predicted outcome
predicted_probs <- predict(stepwise,newdata=test,type='response')
predicted_results <- ifelse(predicted_probs > 0.5,"W","L")
predicted_results <- factor(predicted_results,levels = c("L","W"), ordered = FALSE)
table(test$FXResult, predicted_results)
misClasificError_pred <- mean(predicted_results != test$FXResult)
print(paste('Accuracy of logistic regression model (test dataset):',1-misClasificError_pred))
confusionMatrix(predicted_results, test$FXResult, positive="W")
#Create a ROC curve
perf.obj_test <- prediction(predictions=predicted_probs, labels=test$FXResult)
# Get data for ROC curve
roc.obj_test <- performance(perf.obj_test, measure="tpr", x.measure="fpr")
plot(roc.obj_test,
main="Classifier 1 (test) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc_test <- performance(perf.obj_test, measure = "auc")
# now converting S4 class to vector
auc_test <- unlist(slot(auc_test, "y.values"))
# adding min and max ROC AUC to the center of the plot
maxauc_test<-max(round(auc_test, digits = 2))
maxauct_test <- paste(c("AUC = "),maxauc_test,sep="")
legend(0.37,0.34, maxauct_test,border="white",cex=1.33,box.col = "white")
###Model optimisation###
# Assess the deviance of variables to evaluate the use/non-use of a variable into the model (goodness-of-
fit)
stepwise.anova <-anova(stepwise, test="Chisq")
#Update the model using the variables that are statistically significant (Pr(>|z|)<=0.05) and lower the
deviance significantly (Pr(>Chi)<=0.05)
qualified_variables <-
names(stepwise$coefficients[which(tidy_stepwise$term[which(tidy_stepwise$p.value<0.05)] %in%
rownames(stepwise.anova)[which(stepwise.anova$`Pr(>Chi)`<0.05)])]) #eliminated variables
updated_formula <- as.formula(paste(".~", paste(qualified_variables, collapse="+")))
model_updated <-update(stepwise, updated_formula)
#compare the two models running a chi-squared test
models.anova=anova(model_updated, stepwise, test="Chisq")
#save model parameters in a .csv file
tidy_stepwise<- tidy(stepwise)
write.csv(tidy_stepwise,"Model 1.csv",row.names=FALSE)
#Save odds ratios into a file
odd_ratios_stepwise<-data.frame(c(round(exp(stepwise$coefficients),4)))
lower.ci <- round(odd_ratios_stepwise*exp(-1.96*tidy_stepwise$std.error),4)
upper.ci<- round(odd_ratios_stepwise*exp(1.96*tidy_stepwise$std.error),4)
odd_ratios_stepwise <-cbind(odd_ratios_stepwise,lower.ci, upper.ci)
colnames(odd_ratios_stepwise) <- c("OR", "Lower 95% CI", "Upper 95% CI")
```

```
write.csv(odd_ratios_stepwise,"Odds ratios (Classifier 1).csv",row.names=TRUE)
# Evaluate the model using the training set
fitted_probs <- predict(model_updated,newdata=train,type='response')
fitted_results <- ifelse(fitted_probs > 0.5,"W","L")
fitted_results <- factor(fitted_results,levels = c("L","W"), ordered = FALSE)
misClasificError_train <- mean(fitted_results != train$FXResult)
confusionMatrix(fitted_results, train$FXResult, positive="W")
# Creating performance object
#setting visual parameters
par(mar=c(5,5,2,2),xaxs = "i",yaxs = "i",cex.axis=1.3,cex.lab=1.4)
par(mfrow=c(1,2)) #Illustrate 2 plots side by side
# calculating the values for ROC curve
perf.obj_train <- prediction(predictions=fitted_probs, labels=train$FXResult)
# Get data for ROC curve
roc.obj_train <- performance(perf.obj_train, measure="tpr", x.measure="fpr")
plot(roc.obj_train,
main="Classifier 2 (train) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc <- performance(perf.obj_train, measure = "auc")
auc <- auc@y.values[[1]]
# adding ROC AUC to the center of the plot
maxauc<-max(round(auc, digits = 2))
maxauct <- paste(c("AUC = "),maxauc,sep="")
legend(0.37,0.34, c(maxauct),border="white",cex=1.33,box.col = "white")
#save model parameters in a .csv file
tidy_model_updated<- tidy(model_updated)
write.csv(tidy_model_updated,"Model 2.csv",row.names=FALSE)
#Save odds ratios into a file
odd_ratios_model_updated<-data.frame(c(round(exp(model_updated$coefficients),4)))
lower.ci <- round(odd_ratios_model_updated*exp(-1.96*tidy_model_updated$std.error),4)
upper.ci<- round(odd_ratios_model_updated*exp(1.96*tidy_model_updated$std.error),4)
write.csv(odd_ratios_model_updated,"Table of Odds ratio (Classifier 2).csv",row.names=TRUE)
#Evaluate the model using the test set
#Predict likelihood of outcome = "W" and convert them to the predicted outcome
predicted_probs <- predict(model_updated,newdata=test,type='response')
predicted_results <- ifelse(predicted_probs > 0.5,"W","L")
predicted_results <- factor(predicted_results,levels = c("L","W"), ordered = FALSE)
table(test$FXResult, predicted_results)
misClasificError_pred <- mean(predicted_results != test$FXResult)
print(paste('Accuracy of logistic regression model (test dataset):',1-misClasificError_pred))
confusionMatrix(predicted_results, test$FXResult, positive="W")
#Create a ROC curve
perf.obj_test <- prediction(predictions=predicted_probs, labels=test$FXResult)
# Get data for ROC curve
roc.obj_test <- performance(perf.obj_test, measure="tpr", x.measure="fpr")
plot(roc.obj_test,
main="Classifier 2 (test) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc_test <- performance(perf.obj_test, measure = "auc")
# now converting S4 class to vector
auc_test <- unlist(slot(auc_test, "y.values"))
# adding ROC AUC to the center of the plot
maxauc_test<-max(round(auc_test, digits = 2))
maxauct_test <- paste(c("AUC = "),maxauc_test,sep="")
legend(0.37,0.34, maxauct_test,border="white",cex=1.33,box.col = "white")
#Save odds ratios into a file
odd_ratios<-data.frame(c(round(exp(model_updated$coefficients),4)))
write.csv(odd_ratios,"Odds ratios (Classifier 2).csv",row.names=TRUE)
#Save the model for future use (apply to a dataset of unseen observations)
saveRDS(model_updated, file = "lg2.RDS")
```

Script 2 – Model Update and Prediction version 1.6.R

```
#Model Update and Prediction version 1.6
#Author: Sokratis-Dimitrios Chronopoulos
#Student ID: 201755818
# Input load. Please do not change # This dataset is automatically created by Power BI
`dataset` = read.csv('C:/Users/soc_x/REditorWrapper_2329782c-7da8-484a-97cc-127b342b25be/input_df_fb48a2b1-
015b-41e2-9bff-826ac10eece5.csv', check.names = FALSE, encoding = "UTF-8", blank.lines.skip = FALSE);
# Original Script. Please update your script content here and once completed copy below section back to the
original editing window #
#Check if used packages have been installed in this R session
list.of.packages <- c("dplyr", "plyr", "reshape", "Amelia", "caTools", "e1071", "caret", "tidyverse")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages, force=FALSE)
library(dplyr)
library(plyr)
library(Amelia) # for missing values map
library(reshape)
library(caTools) # for sample.split
library(e1071)
library(caret)
library(tidyverse)
library(broom)
# Function for replacing NAs with modes
Mode <- function (x) {
xtab <- table(x)
xmode <- as.numeric(names(which(xtab == max(xtab))))
if (length(xmode) > 1) xmode <- ceiling(mean(xmode)) #if there are more than one mode, then return the
average
return(xmode)
}
#Setting the current directory as working directory
path <- rstudioapi::getActiveDocumentContext()$path
setwd(dirname(path))
###### Exploratory Data Analysis #######
#Produce a map with missing values vs observed in raw dataset
missmap(dataset, main = "Missing values vs observed")
#calculate NA values proportion per variable from dataframe (raw data): True=NAs, False=Non-NAs
for (i in 1:(length(dataset[-(1:5)]))){
temp <- plyr::count(is.na(dataset[i+5]))
temp$freq<-(temp$freq/nrow(dataset))*100
print(temp)
}
#Create a new dataframe composed of the events occured for each team per fixture
attach(dataset)
unique_features <- as.data.frame(unique(data.frame(FXID,Team,FXHTID,FXATID,HTFTSC,ATFTSC,FXHTResult,
FXATResult)))
detach(dataset)
#Create and populate a new column called FXResult
unique_features$FXResult<-factor(NA,levels = c("L","W"), ordered = FALSE)
for (fixture in unique_features$FXID){
for (team in unique_features$Team[unique_features$FXID==fixture]){
if (Reduce("&",lapply(strsplit(team," ")[[1]],grepl,unique_features$FXHTID[unique_features$FXID==fixture &
unique_features$Team==team]))){
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]<-
unique_features$FXHTResult[unique_features$FXID==fixture & unique_features$Team==team]
}
else if (Reduce("&",lapply(strsplit(team,"
")[[1]],grepl,unique_features$FXATID[unique_features$FXID==fixture & unique_features$Team==team]))){
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]<-
unique_features$FXATResult[unique_features$FXID==fixture & unique_features$Team==team]
}
}
}
#Print the proportions of W/L in terms of aggregation and Home and Away Teams Performance Ratio on the
given dataset
" for testing purpose / future use : inserts FXResult into original dataset
dataset$FXResult<-factor(NA,levels = c('L','W'), ordered = FALSE)
for (fixture in unique(dataset$FXID)){
for (team in unique(dataset$Team[dataset$FXID==fixture])){
dataset$FXResult[dataset$FXID==fixture&dataset$Team==team]<-
unique_features$FXResult[unique_features$FXID==fixture&unique_features$Team==team]
}}"
"Create 6 levels with counts of different combinations occuring from the following variables from each team
of the occured fixtures:
Level 1: count(Action Name, Action Type, Action Result, Qualifier 3, Qualifier 4, Qualifier 5)
Level 2: count(Action Name, Action Type, Action Result, Qualifier 3, Qualifier 4)
Level 3: count(Action Name, Action Type, Action Result, Qualifier 3)
Level 4: count(Action Name, Action Type, Action Result)
Level 5: count(Action Name, Action Type)
Level 6: count(Action Name)
"
#level1<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`', '`Qualifier 4`', '`Qualifier 5`'))
#level2<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`', '`Qualifier 4`'))
```

```
#level3<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`','`Qualifier 3`'))
level4<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`', '`Action
Result`'))
level5<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`', '`Action Type`'))
level6<-plyr::count(dataset, vars=c('FXID','FXHTID','Team','`Action Name`'))
#Now we are going to transform these matrices and merge the features to receive unique values of these
combinations
#level1_final <- cast(level1,
FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3+Qualifier.4+Qualifier.5)
#level2_final <- cast(level2,
FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3+Qualifier.4)
#level3_final <- cast(level3, FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result+Qualifier.3)
level4_final <- cast(level4, FXID+FXHTID+Team~Action.Name+Action.Type+Action.Result)
level5_final <- cast(level5, FXID+FXHTID+Team~Action.Name+Action.Type)
level6_final <- cast(level6, FXID+FXHTID+Team~Action.Name)
#levels_list <- list(level1_final,level2_final,level3_final,level4_final,level5_final,level6_final)
levels_list <- list(level4_final,level5_final,level6_final)
#Now, we can safely remove the individual dataframes we have created in prior to finalise the dataframe
will be used for producing the correlations
rm(level1,level2,level3,level4,level5,level6)
rm(level1_final,level2_final,level3_final,level4_final,level5_final,level6_final)
#Here, we create the final dataframe named global_table that includes all the instances occurring for each
team during the selected fixtures
global_table <- levels_list[[1]] #Fill in the table with the first level according to the order that levels
have been inserted into teh list
for (i in 2:length(levels_list)) # Continue the fullfilment using a for-loop for the remaining levels
{
global_table <- merge(global_table,levels_list[[i]],by= c('FXID','FXHTID','Team'))
}
#calculate NA values proportion per variable level from dataframe (global_table): True=NAs, False=Non-NAs
for (i in 1:(length(global_table[-(1:3)]))){
temp <- plyr::count(is.na(global_table[i+3]))
temp$freq<-(temp$freq/nrow(global_table))*100
print(temp)
}
#Create the dataframe with the tranformed data used to make the Machine Learning Model
global_table$FXResult<-factor(NA,levels = c("L","W"), ordered = FALSE)
for (fixture in global_table$FXID){
for (team in global_table$Team[global_table$FXID==fixture]){
if (Reduce("&",lapply(strsplit(team," ")[[1]],grepl,unique_features$FXHTID[unique_features$FXID==fixture &
unique_features$Team==team]))){
global_table$FXResult[global_table$FXID==fixture & global_table$Team==team]<-
as.factor(unique_features$FXHTResult[unique_features$FXID==fixture])
}
else if (Reduce("&",lapply(strsplit(team,"
")[[1]],grepl,unique_features$FXATID[unique_features$FXID==fixture & unique_features$Team==team]))){
global_table$FXResult[global_table$FXID==fixture & global_table$Team==team]<-
as.factor(unique_features$FXATResult[unique_features$FXID==fixture])
}
}
}
write.csv(global_table,"Global Table (Unforseen data).csv",row.names=FALSE) #Export the dataframe to a .csv
file
#Create a reduced global table removing all columns and rows including NAs in more than 40%
g_other<- global_table[, colSums(is.na(global_table)) < 0.4*(nrow(global_table)-1)] #columns
g_other<-g_other[!(rowSums(is.na(g_other))/ncol(g_other)>0.4),]
#Remove redudant columns
#(The 3-variable columns which end with 'NA' are equal to the 2-variable columns
#that share the first two variables)
g_other<-select(g_other, -ends_with("NA"))
#Remove non-numeric columns (e.g., FXID)
g_other<-g_other[-(1:3)]
# Cleaning NAs using central tendency: Mode
for (var in 1:ncol(g_other)) {
g_other[is.na(g_other[,var]),var] <- Mode(g_other[,var])
}
#### Predictions using the first classifier ####
#Load the created model from previous training
classifier<-readRDS(file = "lg1.RDS")
#Update the model using the variables that exist in both already created model and the new dataset (after
cleaning and preprocessing)
deleted_columns <-names(classifier$coefficients[which(!names(classifier$model) %in% names(g_other))])
#eliminated variables
if( length(deleted_columns)!= 0 ) #Check if there is no variable removal from the model
{
updated_formula <- as.formula(paste(".~. -", paste(deleted_columns, collapse="-"))) #create the updated
formula removing the elimanated variables from the model
}
model_updated <- update(classifier, updated_formula, data=model.frame(classifier)) #update the model with
the updated formula
# Evaluate the model using the predicting set
pred_probs <- predict(model_updated,newdata=g_other,type='response') #the probability of a observation
belongs to "W"
pred_results <- ifelse(pred_probs > 0.5,"W","L") #apply the threshold to obtain the predicted class
pred_results <- factor(pred_results,levels = c("L","W"), ordered = FALSE)
table(g_other$FXResult, pred_results) #confusion matrix
misClasificError_g_other <- mean(pred_results != g_other$FXResult)
```

```
print(paste('Accuracy of logistic regression model (g_other dataset):',1-misClasificError_g_other))
confusionMatrix(pred_results, g_other$FXResult, positive="W")
# Creating performance object (sources:http://scaryscientist.blogspot.com/2016/03/roc-receiver-operating-
characteristics.html, https://github.com/chupvl/R_scripts/blob/master/rocr_wLegend.R accessed: 14/03/2019)
library("ROCR")
#setting visual parameters
par(mar=c(5,5,2,2),xaxs = "i",yaxs = "i",cex.axis=1.3,cex.lab=1.4)
par(mfrow=c(1,1)) # displays two plots side by side
perf.obj_g_other <- prediction(predictions=pred_probs,labels=g_other$FXResult)
# Get data for ROC curve
roc.obj_g_other <- performance(perf.obj_g_other, measure="tpr", x.measure="fpr")
plot(roc.obj_g_other,
main="Classifier 1 (Prediction) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc <- performance(perf.obj_g_other, measure = "auc")
auc <- auc@y.values[[1]]
# adding ROC AUC to the center of the plot
maxauc<-max(round(auc, digits = 2))
maxauct <- paste(c("AUC = "),maxauc,sep="")
legend(0.37,0.34, c(maxauct),border="white",cex=1.33,box.col = "white")
#Save the model for reference
saveRDS(model_updated, file = "lg1_modified.RDS")
#save model parameters in a .csv file
tidy_model_updated<- tidy(model_updated)
write.csv(tidy_model_updated,"Model 1 (Modified).csv",row.names=FALSE)
#### Predictions using the second (enhanced) classifier ####
classifier2<-readRDS(file = "lg2.RDS")
#Update the model using the variables that exist in both already created model and the new dataset (after
cleaning and preprocessing)
deleted_columns <-names(classifier2$coefficients[which(!names(classifier2$model) %in% names(g_other))])
#eliminated variables
if( length(deleted_columns)!= 0 ) #Check if there is no variable removal from the model
{
updated_formula <- as.formula(paste(".~. -", paste(deleted_columns, collapse="-"))) #create the updated
formula removing the elimanated variables from the model
}
model_updated <- update(classifier2, updated_formula, data=model.frame(classifier2)) #update the model with
the updated formula
# Evaluate the model using the predicting set
pred_probs <- predict(model_updated,newdata=g_other,type='response') #the probability of a observation
belongs to "W"
pred_results <- ifelse(pred_probs > 0.5,"W","L") #apply the threshold to obtain the predicted class
pred_results <- factor(pred_results,levels = c("L","W"), ordered = FALSE)
table(g_other$FXResult, pred_results) #confusion matrix
misClasificError_g_other <- mean(pred_results != g_other$FXResult)
print(paste('Accuracy of logistic regression model (g_other dataset):',1-misClasificError_g_other))
confusionMatrix(pred_results, g_other$FXResult, positive="W")
# Creating performance object
perf.obj_g_other <- prediction(predictions=pred_probs,labels=g_other$FXResult)
# Get data for ROC curve
roc.obj_g_other <- performance(perf.obj_g_other, measure="tpr", x.measure="fpr")
plot(roc.obj_g_other,
main="Classifier 2 (Prediction) - ROC Curve",
xlab="1 - Specificity: False Positive Rate",
ylab="Sensitivity: True Positive Rate",
col="blue")
abline(0,1,col="grey")
# Get AUC value from ROC curve
auc <- performance(perf.obj_g_other, measure = "auc")
auc <- auc@y.values[[1]]
# adding ROC AUC to the center of the plot
maxauc<-max(round(auc, digits = 2))
maxauct <- paste(c("AUC = "),maxauc,sep="")
legend(0.37,0.34, c(maxauct),border="white",cex=1.33,box.col = "white")
#Save the model for reference
saveRDS(model_updated, file = "lg2_modified.RDS")
#save model parameters in a .csv file
tidy_model_updated<- tidy(model_updated)
write.csv(tidy_model_updated,"Model 2 (Modified).csv",row.names=FALSE)
saveRDS(model_updated, file = "lg2_modified.RDS")
```

References

1. Aldrich, J., 1997. R. A. Fisher and the Making of Maximum. *Statistical Science,* 12(3), pp. 162-176.

2. Amazon Web Services, n.d. *Model Fit: Underfitting vs. Overfitting.* [Online] Available at: https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html [Accessed 20 March 2019].

3. Analytics Vidhya, 2016. *analyticsvidhya.com.* [Online] Available at: http://i0.wp.com/www.analyticsvidhya.com/wp-content/uploads/2016/03/cost1.png?resize=564%2C232 [Accessed 14 March 2019].

4. Belsley, D. A., Kuh, E. & Welsch, R. E., 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.* New York: Wiley.

5. Bishop, L. & Barnes, A., 2013. Performance indicators that discriminate winning and losing in the knockout stages of the 2011 Rugby World Cup. *International Journal of Performance Analysis in Sport,* pp. 149-159.

6. Burnham, K. P. & Anderson, D. R., 2004. Multimodel inference: understanding AIC and BIC in Model Selection. *Sociological Methods & Research,* Volume 33, pp. 261-304.

7. Duignan, B., 2015. *Encyclopaedia Britannica - The Occam's Razor | Origin, Examples & Facts.* [Online] Available at: https://www.britannica.com/topic/Occams-razor [Accessed 17 March 2019].

8. Fletcher, R. H. & Fletcher, S. W., 2005. *Clinical Epidemiology: The Essentials.* 4th ed. Baltimore: Lippincott Williams & Wilkins.

9. Franks, I., 1993. The effects of experience on the detection and location of performance. *Research Quarterly for Exercise and Sport,* pp. 64: 227-231.

10. Franks, I. & Miller, G., 1991. Training coaches to observe and remember. *Journal of Sports Sciences,* Volume 9, pp. 285-297.

11. Gorakala, S. K., 2017. *R-bloggers.* [Online] Available at: https://www.r-bloggers.com/data-analysis-steps/ [Accessed November 2017].

12. Higham, D. G., Hopkins, W. G., Pyne, D. B. & Anson, J. M., 2014. Performance Indicators Related to Points Scoring and Winning in International Rugby Sevens. *Journal of Sports Science and Medicine,* Volume 13, pp. 358-364.

13. Hughes, A., Barnes, A., Churchill, S. & Stone, J., 2017. Performance indicators that discriminate winning and losing in elite men's and women's Rugby Union. *International Journal of Performance Analysis in Sport,* Volume 17, pp. 1-11.

14. IMS Proschool, n.d. *proschoolonline.com.* [Online] Available at: https://www.proschoolonline.com/blog/data-storytelling-critical-skill-data-scientist [Accessed 17 March 2019].

15. Jones, N. M., Mellalieu, S. D. & James, N., 2004. Team performance indicators as a function of winning and losing in rugby union.. *Journal of Performance Analysis in Sport,* pp. 61-71.

16. Kumar, G., 2013. *(PDF) Machine Learning for Soccer Analytics.* [Online] Available at: https://www.researchgate.net/publication/257048220_Machine_Learning_for_Soccer_Analytics [Accessed 14 March 2019].

17. Laird, P. & Waters, L., 2008. Eye-witness recollection of sports coaches. *International Journal of Performance Analysis of Sport,* pp. 76-84.

18. Landis, J. R. & Koch, G. G., 1977. The measurement of observer agreement for categorical data. *Biometrics*, pp. 159-174.

19. Microsoft Corporation, 2019. *Creating R visuals in the Power BI service - Known Limitations.* [Online] Available at: https://docs.microsoft.com/en-us/power-bi/visuals/service-r-visuals [Accessed 18 March 2019].

20. Morrison, A., 2016. *Risk Ranking Fraud Alerts - Alternative Methods to Logistic Regression,* Glasgow: University of Strathclyde.

21. Nelder, J. & Wedderburn, R., 1972. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General),* 135(3), pp. 370-384.

22. Ortega, E., 2, D. V. & Palao, J. M., 2009. Differences in game statistics between winning and losing rugby teams in the Six Nations Tournament. *Journal of Sports Science and Medicine,* Volume 8, pp. 523-527.

23. Parmar, N., James, N., Hearne, G. & Jones, B., 2018a. Using principal component analysis to develop performance indicators in professional rugby league. *International Journal of Performance Analysis in Sport,* 18(6), pp. 938-949.

24. Parmar, N. et al., 2018b. Team performance indicators that predict match outcome and points difference in professional rugby league. *International Journal of Performance Analysis in Sport,* Volume 17.

25. Shetty, B., 2018. *Towards Data Science - Supervised Machine Learning: Classification.* [Online]
Available at: https://towardsdatascience.com/supervised-machine-learning-classification-5e685fe18a6d
[Accessed 17 March 2019].

26. Statistics Canada, 2017. *Calculating the mode.* [Online]
Available at: https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch11/mode/5214873-eng.htm
[Accessed March 17 2019].

27. Stein, M. et al., 2017. How to Make Sense of Team Sport Data: From Acquisition to Data Modeling and Research Aspects. *Data*.

28. Vaz, L., A. M., Carreras, D. & Morente, H., 2011. The importance of rugby game related statistics to discriminate winners and losers at the elite level competitions in close and balanced games. *International Journal of Performance Analysis in Sport,* pp. 130-141.

29. Vrieze, S. I., 2012. Model selection and psychological theory: a discussion of the differences between the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). *Psychological Methods,* Volume 17, pp. 228-243.

30. Walker, M., 2013. *Data Science Central.com - Batch vs. Real Time Data Processing.* [Online]
Available at: https://www.datasciencecentral.com/profiles/blogs/batch-vs-real-time-data-processing
[Accessed March 22 2019].

31. Watson, N., Durbach, I., Hendricks, S. & Stewart, T., 2017. On the validity of team performance indicators in rugby union. *INTERNATIONAL JOURNAL OF PERFORMANCE ANALYSIS IN SPORT,* Volume 17, pp. 609-621.

32. Wikipedia contributors, 2018. *Classification rule,* s.l.: Wikipedia, The Free Encyclopedia..

33. Wikipedia contributors, 2019. *Statistical classification.* s.l.:Wikipedia, The Free Encyclopedia.

34. Wikipedia contributors, n.d. *Wikipedia - Cross-validation (statistics),* s.l.: s.n.

35. Woods, C. T., Sinclair, W. & Robertson, S., 2017. Explaining match outcome and ladder position in the National Rugby League using team performance indicators. *Journal of Science and Medicine in Sport,* Volume 20, pp. 1107-1111.

36. Yang, Y., 2005. Can the strengths of AIC and BIC be shared?. *Biometrica,* 92(4), pp. 937-950.

37. Zhao, S., 2017. *Experian Data Quality.* [Online] Available at: https://www.edq.com/blog/what-is-etl-extract-transform-load/ [Accessed 15 March 2019].