What Does It Take To Win: Predicting Performance in Cycling Based on Training Load

Laura Kooijman
: u222718

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Hendrickson
Jung

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
January 2021

Preface

The last months I have been researching the topic of Training Load in endurance sports with great pleasure and it has become clear to me that the last few years the field has made big steps in applying data science to the sport. I would be very grateful if my research would be make a contribution to this development. I would like to thank Teun van Erp for his guidance the past few months and for providing me with the data and I would like to thank Drew Hendrickson for his advice in our weekly meetings, it helped me look at things from multiple perspectives.

I hope you enjoy the reading.

Laura Kooijman

What Does It Take To Win: Predicting Performance in Cycling Based on Training Load

Laura Kooijman

In professional cycling training schedules are optimized to perfection. But to know on beforehand if the training schedule has the desired effect, there is the need to know what effect the training had on race performance. In this research a logistic regression, a support vector machine and a random forest are developed to predict race performance of a professional female cyclist, based on training load. The data consists of the races and training of 2017-2019. The research question that will be answered is: To what extent can race performance be predicted in cycling, based on training load?

Athlete data is often limited in size as athletes only can do a number of races per year which makes the data impractical for predictive modelling. This study investigates which techniques are helpful in classifying race performance. Class balancing is performed using weight adjustment and the SMOTE technique. In addition to that, PCA is performed. The random forest with weight adjustment gave the best result with a F1-score of 0.88, which shows that it is possible to predict race performance with a small dataset. The PCA showed an improvement in prediction for the SVM with an F1-score of 0.872, which is an improvement but not as high as the random forest. This means that the PCA was not beneficial for this dataset.

1. Introduction

When the Tour the France started in 1903, it was a small race with just 60 cyclists competing. Their bikes were from titanium or aluminium, they were wearing their everyday clothes and they had to be self-sufficient. Nowadays, bikes are made from carbon, clothes are designed to be light-weight and a whole crew is working behind the cyclist to optimize their performance. The only thing that has stayed the same over the years is the mentality to win. One of the aspects that is important for athletes to optimize, is their training plan. In training, they face the trade-off of training too hard and risk injury or overtraining, or training too little, and having a competitor that is better. A measure that is used to show this trade-off is Training Load (TL). In an ideal situation, the Training Load is maximized in the training period, and minimized in the week before the race, to be able to perform well at races.

Researchers have attempted to model the relationship between Training Load and performance in ice skating (Orie et al. 2020) as well as in triathlon (Stoeber, Uphill, and Hotham 2009) but for cycling, this specific relationship is quite understudied as it is hard to define performance in a sport like cycling. In ice skating, it is simple, the format is often the same and the environment much more controllable, whereas in cycling, races do not have a rigid format, which makes it much harder to compare performance in different races. Nevertheless, it is interesting to see to what extend it is possible to predict race performance as on training load.

The prediction of this relationship would not only be interesting for athletes and their coaches, but also for researchers in the field of data mining. The data provided by cyclists is often of good quality but of a small size. This makes it hard to generalize results and to train models effectively on the data. This study will show if it is possible to train simple linear models on a very small dataset. The focus will be on using the data as effectively as possible to make accurate predictions. The results of this study could be used in the future as a guide on what techniques and algorithms work and do not work for small datasets.

In this study, training and racing data of a professional female cyclist will be used to develop a model that is able to predict race performance based on training data. The race performance is predicted on the basis of classification, the race results are divided in two categories; a top-10 result or lower than top-10 result. To come to these results, the following questions are formulated:

The main question is: *To what extent can race performance be predicted in cycling, based on training load?*

This question is substantiated by the following sub-questions:

- RQ1 - Which model is best suited when evaluated by the F1-score? The models that are tested are Linear Support Vector Machine, Random Forest or Logistic Regression.
- RQ2 - What is the effect of balancing the classes and which method is best suited for that? The methods that are tested are weight adjustment and Synthetic Minority Oversampling Technique (SMOTE)
- RQ3 - To what extend does Principal Component Analysis contribute to the performance of the models?
- RQ4 - Which features are important in the prediction of performance?

The sub-questions are designed to be able to evaluate what the effect is of each method on the data and so the be able to find the optimal combinations of techniques for a small dataset. All model performances are evaluated by the F1-score. The first question will find which predictor is best suited for this dataset. Secondly, different balancing techniques are tested to see if balancing the classes has any effect on the performance of the models. Unbalanced classes often lead to models that classify every instance as the majority class as that gives a fair result on average. Balancing methods try to correct for this behaviour, which will eventually lead to more realistic outcome. Thirdly, PCA is used as it is known to reduce dimensionality and to reduce noise. In addition to that, the dataset that is used in this thesis has a lot of correlations between the features and PCA will remove those correlations. Lastly, the important features will be selected from the model that performs best. This selection is done to give a guideline to athletes and their coaches because it shows them which training variables they have to focus on.

The following section gives an overview of literature that is available on these concepts. Secondly, the data and methodology are discussed, which will be concluded with the results of the models and the discussion of those results. Finally, answers to the research questions will be provided.

2. Related Work

In this part of the thesis, a summary of the relevant literature is presented. In the first part, relevant literature regarding Training Load is discussed. In the second part, relevant literature regarding machine learning techniques for similar situations as this one are discussed.

2.1 Training Load

To evaluate what the effect of training is on the body, training needs to be quantified. Training can be quantified with data such as duration (minutes), intensity (heartrate) or power output (watts). Trainig Load is a parameter that measures the impact of a training session on an athlete. It can be expressed in different ways such as session Rate-of-Perceived-Exhaustion (sRPE), mean heart rate (HR) and Power Output (PO) (van Erp 2020). sRPE is a subjective measure based on the Borg Scale (Borg 1998). It is a scale ranging from 6-20 which represents how the athlete perceived the intensity of a workout, the rate of perceived exertion (RPE). Athletes rate their workouts with the Borg Scale as soon as they finish. For the sRPE, the RPE is multiplied by the duration of the workout in minutes. Mean heart rate is the simply the average heart rate during the session and the Power Output is the average power in watts athletes produce during a workout.

It is important to know if a certain training measure can be used to quantify TL before starting an analysis. All of these previously mentioned training measures are correlated with each other and are proved to be valid measures for TL in training (Sanders D 2017). Van Erp and De Koning (2018) investigated if these measures behave the same in both training and racing, which is the case. Their research makes it possible to use TL for both training and racing and thus quantifying both efforts. Therefore, TL in the form of sRPE is used as input for the prediction model of this thesis.

2.2 Determinants of Cycling Performance

To provide some background on the relationship between training load and race performance in cycling, it is good to know which variables are important in determining cycling performance. Philips and Hopkins (2020) did a systematic review on the determinants of cycling performance. They divided the determinants of cycling performance in four dimensions; features related to the individual cyclist, tactical features, strategic features and global features. Training Load belongs to the features related to the individual cyclist, more specifically, related to training techniques. The study shows that an improvement in training techniques shows an improvement in race performance (Philips and Hopkins 2020). Secondly, the study shows that the features are intertwined and that there is a complex interplay between the different features and dimensions, and that it is therefore difficult to understand the relation between the features. This

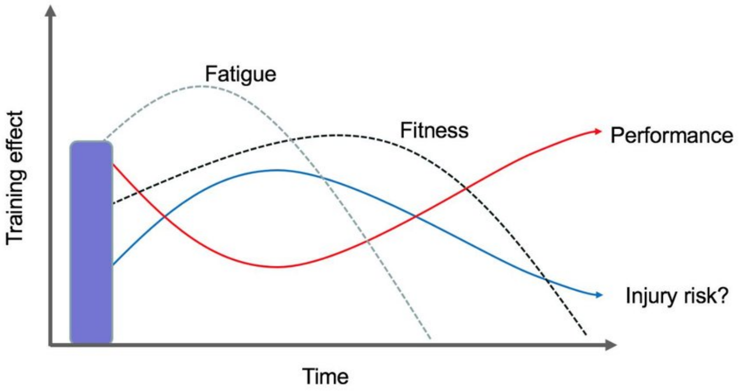

Figure 1
The Relationship between Fitness and Fatigue in Relation to Performance and Injury Risk (Calvert et al. 1976)

finding is something to keep in mind when modelling the relationship between TL and performance.

Interestingly, the authors state that improvements in training techniques has reached a ceiling. If this study shows that race performance can be predicted by Training Load, it indicates that there are still chances to improve training techniques. This is a sign that the thesis can be a useful contribution to the field.

2.3 Sports Analytics for Professional Speed Skating

Another study that provides helpful insights in what is already done in the sports industry is the article of Orie et al. (2020). In this study, the data of a professional speed skater is analysed using LASSO Regression. What is especially helpful is the visualization of the relation between training load and performance which is shown in Figure 1. Orie et al. (2020) state: "When doing a certain training routine, it can be expected that the relationship is in fact curved, with peak performance being achieved at a certain optimal load on the human body. Doing too little will not achieve the right effect, but doing too much of the training also produces sub-optimal performance. Specifically, one can expect thresholds in the training load above (or below) which performance will rapidly diminish.". This quote corresponds with the figure.

In addition to that, this study faces a similar challenge with regard to the time sensitivity of the data. Orie et al. spend a considerable amount of effort into feature construction and they approach time with the Fitness-Fatique model: the effect of a training becomes less as the days pass as well as the fatigue of that training. This phenomenon can be modelled with the following mathematical function:

$$h_{ff}(m) = (e^{-\lambda_{fit}m} - e^{-\lambda_{del}m}) - Ke^{-\lambda_{fat}m}, m \geq 0$$

The values for the parameters are as followed: $\lambda_{fit} = 50$ days, $\lambda_{fat} = 15$ days, $\lambda_{del} = 5$ days, where *fit* determines the positive effect of training, *fat* determines the shape of the fatigue curve and *del* affects the exponential function that influences the fitness (*fit*). As the authors state, the results for the LASSO Regression did not lead to a good fitting model with an R^2 of 0.721, but the time approach of fitness and fatigue and the visualization of the optimum are valuable insights that can be taken into account when designing a model for this thesis.

2.4 Predicting Running Injuries from Training Load

An article that is a good addition to the article of Orie et al. is the article of Dijkhuis et al. (2017). In this article a model which predicts running injuries is developed based on Training Load. They did this using a Random Forest. This article shows, as well as the article of Orie et al., that Training Load has the trade-off in hours of training as can also be seen in Figure 1. Before training the random forest, feature selection (based on random forest as well) was used to identify which variables were predictive for identifying the risk of injuries. Out of 85 variables, eventually 10 were selected. The features that are selected by the model are:

- Average workload week 2
- Sum workload week 2
- Percentage change monotony between week 1 and 2
- Acute:chronic ratio 7 over 42 week 7
- Acute:chronic ratio 7 over 28 week 7
- Percentage change strain between week 1 and 2
- Percentage change workload between week 2 and 3
- Acute:chronic ratio 7 over 42 week 2
- Percentage change strain between week 2 and 3
- Strain 2

The eventual model had an accuracy of 67% (Dijkhuis et al. 2017). The research showed that the feature selection had a positive effect on the accuracy of the model with an improvement in accuracy of 6%.

These findings are interesting because the data and the research design show a lot of similarities with the thesis. A comparison between this set of selected features and the set of features selected in the thesis can be made afterwards to see if the results align. Especially for the time component in both studies it will be interesting to see if there is a time-span, such as four weeks prior, that is more predictive than other time-spans. In addition to that, this study shows that the Acute:Chronic ratio plays an important role in the prediction of injuries. This feature is therefore also included in the thesis to see if it plays a similar role in the prediction of performance.

2.5 Analysis of Dimensionality Reduction Techniques on Big Data

Predictions on classification can be done with various models. Finetuning by means of pre-processing or feature engineering gives even more options. The article of Reddy et al. (2020) provides an overview of testing different models with different dimensionality reduction techniques on different datasets to give a guide on which combination of models has the desired predictive power for a certain dataset. In this study a Decision Tree, Support Vector Machine, Naive Bayes Classifier and a Random Forest are trained on medical datasets of varying sizes. For the dimensionality reduction technique Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are used.

The results show that firstly, the PCA performs better than the LDA in all measures and that the Decision Tree and Random Forest are not much affected by the dimensionality reduction. Secondly, the SVM and Random Forest outperform the other classifiers. Thirdly, the classifiers with PCA performed better than the classifiers without PCA. Fourthly, PCA was more useful for datasets with a high variance and lastly, (Reddy et al. 2020) found that when the size of the dataset is too small, the dimensionality reduction techniques have a negative impact on the classifiers.

These results provide a useful guide for this thesis on multiple aspects. The method of testing different combinations of algorithms and reduction techniques shows how much the performance per combination can vary. The effect of PCA on the SVM and Random Forest provide mixed results and it will be interesting to see what the results of these algorithms will be on the thesis dataset as the data has a high variance but is limited in size.

2.6 The Curse of Class Imbalance and Conflicting Metrics with Machine Learning for Side-channel Evaluations

Before answering RQ2 about balancing classes, it is good to provide some background on the topic. There are several ways of balancing classes such as assigning different weights to the classes in a model, deleting samples of the majority class by undersampling or creating new synthetic samples by over-sampling. The article of Picek et al. (2019) provides an overview of the effectiveness of different balancing techniques. In the study they tested the SMOTE over- and under-sampling technique and adjustments of class weights. The authors tested these techniques on a SVM and a Random Forest. Their results show that the classifiers did not generate reliable results when used on imbalanced datasets without balancing method. Secondly, they found that class-weight adjustment improves performance but SMOTE provides an even better result, even in the presence of noise (Picek et al. 2019). These findings are taken into account with the design of the balancing methods for the thesis.

These selected articles provide a short overview of what has already been done in the field of sports analytics and data science for small datasets. They give a solid base for the upcoming research. All articles show that it is possible to make predictions with a similar type of data as well as a similar size of dataset with both SVM's and Random Forests.

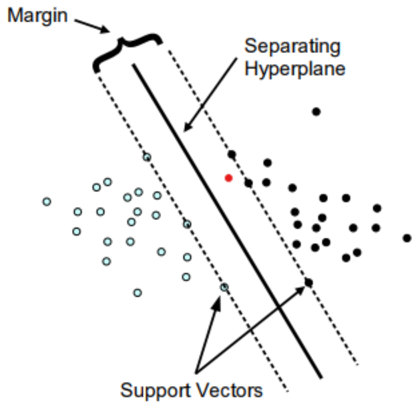

Figure 2
Classification with Linear SVM (Meyer and Wien 2015)

3. Method

This section describes the models and algorithms that are used to answer the research questions. The techniques will be discussed per research question.

3.1 Models

3.1.1 Support Vector Machine. A Support Vector Machine is a relative simple supervised learning algorithm. It makes predictions based on separating a hyperplane by maximizing the space between the two closest data points of two different classes (Meyer and Wien 2015). A graphical visualization of this phenomenon is shown in Figure 2. Data can be linear as well as non-linear as the hyperplanes can take different forms. For this thesis, a linear SVM is used as SVM's are known to be robust classifiers.

3.1.2 Logistic Regression. Logistic Regression does the exact opposite of the SVM as it draws a line as close to the data points as possible. Instead of drawing a line between the classes, Logistic Regression calculates the odds of a data point being one class or the other, based on the sigmoid function which give the log-odds:

$$h_\theta(x) = \frac{1}{1 + e^- \theta^T x}$$

After the log-odds are calculated, a linear relationship is assumed between the log-odds and the predictor variables. For binary classification, this means that when the odds of a class are higher than 0.5, the data point is predicted to be that class (Yildirim 2020).

3.1.3 Random Forest. Where the SVM and Logistic Regression perform their predictions on a linear base, a random forest bases it's outcome on the outcomes of it's decision trees. A random forest is build from decision trees which in turn are build from different leaf nodes. The leaf nodes each perform a logical test on the predictors and give the probability of a data point being one class or the other. These leave nodes together give a weighted decision on the classification of a data point (Yiu 2020). Because decision trees are sensitive to which part of the dataset is fed first, random forest are introduced.

Random forests take the predictions of the individual decision trees into account and therefore protect for their individual error.

These three models are all rather 'simple' supervised learning algorithms that work differently. A simple supervised model is preferred as the dataset is small and an overly complicated model can easily lead to overfitting (Lever, Krzywinski, and Altman 2016).

3.2 Balancing

As the classes in the thesis dataset are imbalanced, there is a need to compensate for that imbalance. When an imbalance dataset is fed into the models, especially the ones on a linear base, the models will predict all data points as the majority class (Khoshgoftaar, Gao, and Seliya 2010). To compensate for the imbalance, there are roughly two techniques; "in model" adjustment by adjustment of weights and modifying the size of the classes. Both techniques are used in this thesis.

3.2.1 Adjustment of Model Weights. The adjustments of class weights in the process of fitting a model is an elegant way of accounting for class imbalance. As a model learns from each data point, it adjusts it's function slightly every time a prediction is not correct. Class weights provide a way to give more emphasis on a class. The weights are applied to the cost function. This means that when a bigger weight is assigned to a class, a wrong prediction is marked as more important than a wrong prediction in the class with a smaller weight (Zhu et al. 2018). The consequence of this is that the model will adapt it's parameter more towards the class with a higher assigned weight. This way, it is still possible to make predictions on a dataset with classes of different sizes.

3.2.2 SMOTE. The Synthetic Minority Oversampling Technique is a technique that over-samples the minority class. The synthetic data points are created using k-nearest neighbour (knn) of the minority class. Along the line of this knn, synthetic data points are randomly chosen, depending on the amount of extra points needed (Chawla et al. 2002). This leads to a more realistic dataset than when the minority class is simply duplicated and therefore leads to a more realistic classification.

3.3 PCA

Principal Component Analysis provides a way of representing the information data holds in a different way. Pearson formulates the analysis as finding "lines and planes of closest fit to systems of points in space" (Wold, Esbensen, and Geladi 1987). In other words, principal components represent the explained variance of the variables. PCA can have many applications such as simplification, dimensionality reduction, visualization and variable selection (Wold, Esbensen, and Geladi 1987). In this thesis, the main goal of PCA will be simplification of the data and dimensionality reduction. As some features share the variance due to the fact that these features contain information of other features (Training Load of 1 week is also present in the Training Load of 8 weeks), PCA could capture this information much more efficient than the traditional way of representing data.

3.4 Feature importance

The feature will be determined by the best performing models. These models each determine feature importance in their own way. For the SVM, feature importance is calculated by the classifier coefficients which are determined by the hyperplanes. The logistic regression does in some sense the same as it also calculates the coefficients of the line drawn through the data points. The random forest calculates importance based on impurity or information gain, dependent on which parameter the model is set. For impurity the features with the lowest value are most important while for information gain, the highest value is important.

4. Experimental Setup

In this section, the data and experimental setup is described. In addition to that, the selected packages and hyper-parameters are described and their choices are motivated.

4.1 Data

The data provided consists of two datasets. Dataset 1 contains the raw data of all training entries and dataset 2 contains a set of calculated features. The dataset with calculated features consists of all race entries of 2017 to 2019 of a professional female cyclist. A list of variables of the original dataset is included in the Appendix Table 8. The entries in dataset 2 represent all races where the cyclist had a key-role in the team and was expected to end in the top 10. The races where she had a supportive role, and was therefore not expected to end in the top 10, are excluded. Dataset 2 is used to train the models on.

Table 1 shows the added variables. The Load-ratio is included because research shows that the Load-ratio is a good predictor for performance (White 2020) (Dijkhuis et al. 2017). The categorical results are added because they will be the predicted value. The race results are divided in a top-10 class (n=58), meaning a good performance and lower than top-10 class (n=21), meaning a bad performance. The division in top-10 results and <top-10 results is made because cyclists often get price money and UCI points in large races when they finish in the top-10, which leads to a better world-ranking. This ranking is eventually determining for the contracts in cycling teams.

Table 1
Added Variables

Name	Data type	Description
Year	Int	Year derived from Indexnr. Possible values: 2017, 2018, 2019.
Loadratio	float	sRPE-1week divided by sRPE-4week.
Race results	int	Race result of the race at that indexnr.
binresult	bool	1 = good performance, 0 = bad performance.

Because outliers play an important role in the PCA, a thorough outlier detection analysis has been performed. Figure 8 in the appendix shows the boxplots of the individual features categorised by the output variable ’Results’. From the boxplots can be seen that half of the variables contain observations that fall outside the whiskers of the boxplot. Some research in the trainingpeak data (the source of the training data) shows that all observations are legitimate training sessions and that although they are extreme, the observations are not a mistake. Therefore, the outliers are preserved in the dataset. Because the sensitivity of PCA to the scaling of the data, it is vital that there are no outliers in the dataset (Aldehim and Wang 2017). To overcome this problem, the data is scaled by using the whiskers of the boxplots. This way, the large value is preserved, but the scaling is more suitable to the data.

After the EDA, the data is subsetted, the final subset is included in Table 2. The final subset consists of all training variables and the binned race results as the to-be predicted value.

4.1.1 Variables. To get an idea of what the variables represent, a short description of their meaning is included in Table 2. In this section however, a more thorough explanation is given. Every observation in the dataset represents a race where the cyclist was expected to obtain a top-10 result. All other variables (except the Loadratio) represent the training done in the weeks before the race. To come to the value of these variables, a weighted average is taken of all training in 7, 28 and 56 days before the race. The sRPE, already mentioned in the Related Works section, represents how hard a training session was using the Borg Scale (Borg 1998). The sRPE-week variables are calculated by taking the weighted average of the sRPE’s multiplied by the minutes of training as the formula states below where t = time in days. Multiplying the sRPE by the training time is a common way to express Training Load.

$$sRPE = \frac{\sum^{t} RPE * duration_{min}}{t}$$

To see if there are any differences between good and bad performances at first sight, a density plot is made of all variables separated by the output variable ’binresult’. Figure 9 in the appendix shows that there is a visual difference in Zone1-4week and sRPE-4week. It will be interesting to see if these variables also play an important part in the prediction of the performance which will be answered in RQ4.

4.2 Experimental Procedure

In this section, the experimental procedure is discussed per research question. The section includes the algorithms used and the motivation behind the (hyper)parameters.

4.2.1 Models. For the first research question, three algorithms are selected to predict cycling performance, namely: Support Vector Machine, Logistic Regression and Random Forest.

The logistic regression is used because it is one of the simplest models, while also being very capable of classifying small datasets. The choice of hyperparameters is as followed: the max iterations is set to 10000, the solver is set to ’liblinear’ and the regularization (penalty) is set to ’l1’. Lasso regression (l1) is used as for regularization as it shrinks the less important features to zero, which is ideal when there are a lot of

Table 2
Final set of Variables

Nr.	Name	Data type	Description
1	Loadratio	float	sRPE-1week divided by sRPE-4week.
2	sRPE-1week	float	Sum of sRPE of all training of one week prior to the race.
3	sRPE-4week	float	Sum of sRPE of all training of four weeks prior to the race.
4	sRPE-8week	float	Sum of sRPE of all training of eight weeks prior to the race.
5	Zone1-1week	float	Sum of minutes ridden in zone 1 intensity one week prior to the race.
6	Zone2-1week	float	Sum of minutes ridden in zone 2 intensity one week prior to the race.
7	Zone3-1week	float	Sum of minutes ridden in zone 3 intensity one week prior to the race.
8	Zone4-1week	float	Sum of minutes ridden in zone 4 intensity one week prior to the race.
9	Zone5-1week	float	Sum of minutes ridden in zone 5 intensity one week prior to the race.
10-14	Zone..-4week	float	Same as one week variable but then sum of 28 days.
15-19	Zone..-8week	float	Same as one week variable but then sum of 56 days.
20	Binresult	bool	1=Race result <= 10, 0=Race result > 10.

features (Nagpal 2020). The L1 regression is in the sklearn package only supported by the liblinear solver, so that choice was clear as well.

In addition to the logistic regression, a (linear) support vector machine is trained. This is done to compare the performance because the logistic regressor has the tendency to overfit. It will be interesting to see how the performance of the SVM is compared to the linear regression as they both work on a linear basis but have their own limitations. An important note on why the SVM is chosen in addition to the logistic regression model is that the SVM will still work when the number of features is larger than the number of observations (sklearn 2020). This shows the robustness of the model when the ratio between observations and features is not optimal.

The final model that is used, is the random forest. In previous studies random forests show their capabilities of classifying small datasets (Shaikhina et al. 2015) (Thanh Noi and Kappas 2018). As a random forest has a several parameters and therefore several tuning possibilities, a GridSearch for the best hyperparameters is performed. The GridSearch eventually only lead to an overfitting model, which is to be expected when the risk of overfitting is rather big with the small dataset with imbalanced classes. Therefore, there is chosen to leave the parameters to the default settings. One exception is the split criterion, which is set to entropy. Entropy is chosen because it outperformed the Gini Impurity.

4.2.2 Balancing Methods. Another component that has to be taken into account is the size of the classes. Since the ’good performance’ has 58 entries and the ’bad performance' has only 21 entries, we are dealing with imbalance. This imbalance could cause the models to have big preference towards the majority class and leads to poor classification of the minority class (Khoshgoftaar, Gao, and Seliya 2010). To answer Research Question 3 and to overcome the problems of class imbalance, the classes are balanced using the following methods: 'class weighing' and the SMOTE method.

For the weight adjustment GridSearch is used to find the optimal balance between the classes. The GridSearch is used on each individual model to see if there is a difference in the optimal class weight between the models. Table 3 shows the values that have been tried in the GridSearch.

For the SMOTE method (Synthetic Oversampling Technique), the imblearn API is used. The SMOTE oversampling method generates synthetic data points of the minority class. This would be beneficial for two issues at once as it gives an even distribution of the classes and at the same time, it increases the size of the dataset.

Nr.	Weights per class
1	0: 100, 1: 1
2	0: 10, 1: 1
3	0: 1, 1: 1
4	0: 1, 1: 10
5	0: 1, 1: 100

Table 3
Weights used in GridSearch for Class Balancing.

4.2.3 Principal Component Analysis. To answer RQ3, PCA is performed. This is done by looking at the explained variance of the transformed components. Before fitting the Principal Components (PC's), the outliers are removed and the data is scaled with StandardScaler. This type of scaling calculates the z-score independently per feature. After the PCA is performed, the analysis of the original models (SVM, logistic regression and random forest) is performed for a second time with the PC's. This is done with three different subsets of components namely, 3 PC's, 6 PC's, 10 PC's and 16 PC's. The four subsets will show what the effect of the transformation on the different algorithms is.

4.2.4 Feature Importance. After the model performances are known, the feature importance is determinate using the 'feature_importance_' option of sklearn of the best performing model.

4.2.5 Evaluation. Before fitting any model, a baseline is established. For this thesis, the majority class is used as a baseline as models tend to get towards the baseline when the model is poorly fitted. For this study, the baseline is (58/79) = 0.734.

$$Baseline = \frac{\#\ observations\ majority\ class}{\#\ total\ observations}$$

The models are evaluated using Leave-one-out (LOO) cross-validation and is optimized with F1-scoring. Although the LOO CV is computationally expensive, it gives the option to maximize the training set and still have a test set as the model is trained and tested with each observation. This evaluation metric is therefore often not possible, but with such a small dataset, it is. The F1-score as evaluation metric is chosen over accuracy because of the class imbalance.

$$F1score = 2 * \frac{precision * recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}(fp + fn)}$$

Accuracy would give a biased view on the results as the accuracy will naturally be high if the model just predict the majority class.as the formula shows, the F1-score is the weighted mean of Recall and Precision and gives a much more nuanced picture of the results.

4.3 Feature Importance

4.4 Implementation

The results are obtained with the following packages:

- Python version 3.9.1
- Scikit-learn version 0.24.0 for all models, the evaluation metrics, weight adjustment, GridSearch and PCA
- Imbalanced-learn version 0.7.0 for SMOTE
- Seaborn version 0.11.1 and Matplotlib version 3.3.3 for visualization of the results

5. Results

This section presents the results obtained by following the methodology listed above. The results are listed per research question and afterwards, answered in the discussion. Finally, the main research question will be answered in the discussion based on the answers of the sub-questions.

5.1 Model Performance

The first RQ states: *Which model is best suited when evaluated by the F1-score?*. The models that are tested are logistic regression, a support vector machine and a random forest. All models are tuned with the hyper-parameters mentioned in the Method section. The outcome of these models is shown in Table 4.

The results in the table show that the random forest achieved the highest F1-score (F1 = 0.85), followed by the logistic regression (F1 = 0.84) and lastly, the SVM (F1 = 0.79). To recall from the Method section, the baseline is 0.734. From this we can conclude that all models perform better than the baseline. To see how the models predict the observations, a confusion matrix is placed in the appendix. The first section of Table 9 shows the predictions of the models in a Confusion Matrix. These results show that the Precision and Recall for the 1-class, the "good performance" class is very good. The Precision and Recall is much lower for the 0-class, the "bad performance" class.

Model	**F1-score**
Logistic Regression	0.84
Linear SVM	0.79
Random Forest	0.85

Table 4
F1-score of simple models

5.2 Balancing Classes

As mentioned earlier, the dataset is imbalanced with 58 observations in the "good performance" class and 21 observations in the "bad performance" class. The results

	F1-score		
Model	**No Balance**	**Weights**	**SMOTE**
Logistic Regression	0.84	0.85	0.68
Linear SVM	0.79	0.85	0.58
Random Forest	0.85	0.88	0.79

Table 5
F1-score of models with weight adjustment and SMOTE

in the previous section showed that the models have a slight preference towards the majority class, which is to be expected. The balancing methods as explained in the method section try to account for this behaviour. Table 5 shows the results for both the class weights and the SMOTE balancing. The models without balance are included as well for comparison.

5.2.1 Weight adjustment. The results show that balancing the classes leads to an increase in performance of all models. The logistic regression and SVM had a F1-score of 0.85 and the random forest had a F1-score of 0.88. The weights that were chosen by GridSearch, giving the highest F1-score were the following:

- Logistic Regression: class 0: 1, class 1: 100
- Support Vector Machine: class 0: 1, class 1: 10 0:1, 1:100 (both provided the same results)
- Random Forest: class 0: 100, class 1: 1

These weights are notable because these weights are almost the opposite of each other. Intuitively, one would expect that the weights would be adjusted towards the smaller class. For the random forest, that is indeed the case, while the SVM and the logistic regression laying even more weight on the majority class increased the F1-score. The abnormality of this choice is confirmed in the Confusion Matrix (Table 9). Before adjustment of the weights, the models already had a preference for the majority class and after adjusting the weights, this preference became even stronger. The eventual F1-score is still higher than the baseline because it takes the mean of the two classes. Unfortunately, it does not give the desired effect of balancing the classes evenly. This is probably be the reason that the random forest without feature selection and with weight adjustment is the best performing model, for this model it is possible to make a better classification of the minority class with the weight adjustment.

5.2.2 SMOTE. Although the SMOTE technique was promising, the results are less impressive. Only the random forest performs better than the baseline with an F1-score of 0.79. The SVM (F1 = 0.58) and the logistic regression (F1 = 0.68) are not able to make better predictions with the addition of the synthetic samples. The random forest is handling the synthetic samples slightly better, although still performing worse than with adjusting weight or without balancing the classes at all. To find the reason of these outcomes, we again look at the confusion matrices.

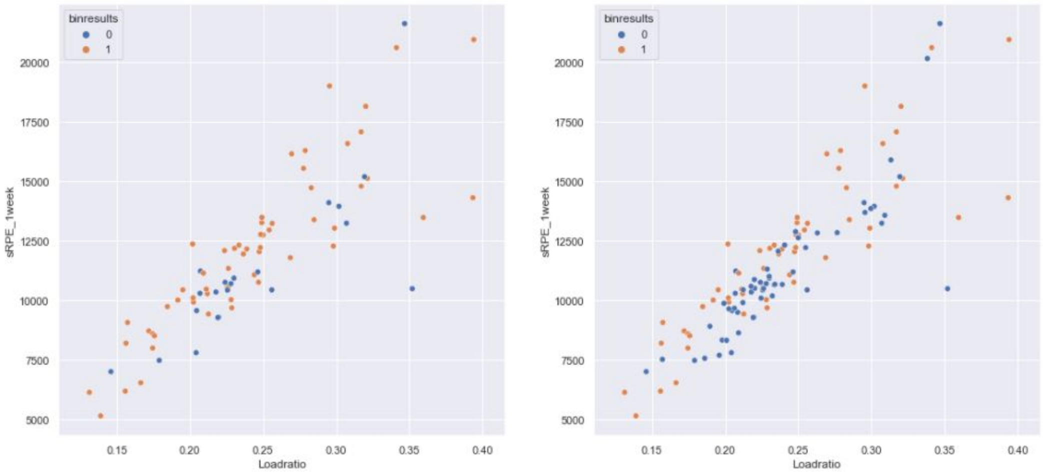

Figure 3
Plot of new SMOTE samples in contrast with original data

Figure 3 shows a scatterplot of two variables of the thesis dataset before and after the SMOTE observations are added. The plot on the right shows that, although centered in the middle, the newly created observations appear logical. Because of the new observations created by SMOTE, there is no majority class anymore. The logistic regression and the SVM are now able to better classify the ’bad performance’ class, which was previously the minority class which can be seen in the Confusion Matrix in the appendix in Table 9. The reason that the F1-score is not higher, is that the classification of the ’top-10’ class, has become more difficult. This leads to an F1-score of both models that is worse than the baseline. For the random forest, the same reasoning applies to some extent, but this model is able to make a much better classification of the ’good performance’ class. Although the SMOTE technique led to a decrease of performance of the models, it might have given a more realistic picture of the model performance by taking away the possibility to just take the majority class. As with the conclusion of the previous RQ, the models would benefit from more data.

5.3 Principal Component Analysis

As the variables share part of their information, the information sRPE-1week contains is also present in sRPE-4week for example, it is expected that the features share variance. Therefore, PCA is performed to get insight in how much variance is shared and if the transformation of features into principal components improves the performance of the models.

Figure 4 shows the variance explained by the principal components. The figure shows the typical asymptotic shape of explained variance with a stabilisation from 13 components. On the basis of the variance explained by the components, different sets of components are chosen to train the models on. For the analyses are 3, 6, 10 and 16 components used to see if there is any difference in performance.

Table 6 shows the F1-score per set of selected components. In addition to the table, figure 5 shows the F1-score for every set of PC’s. Together, the table and the graph give a complete outline of what PCA does for the performance of the models. For the random

	F1-score			
Model	**3 PC's**	**6 PC's**	**10 PC's**	**16 PC's**
Logistic Regression	0.85	0.83	0.84	0.84
Linear SVM	0.85	0.85	0.86	0.87
Random Forest	0.79	0.79	0.82	0.77

Table 6
F1-score of models with different sets of Principal Components

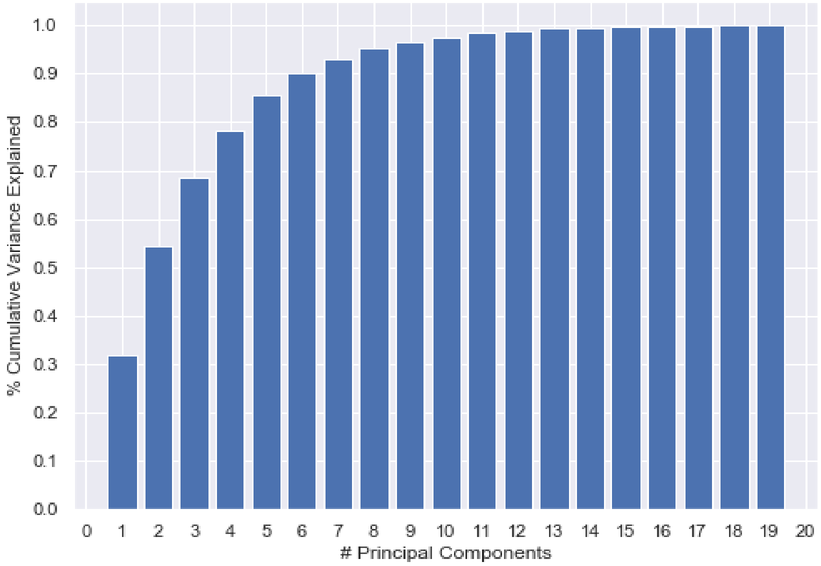

Figure 4
Variance explained by principal components

forest for example, performance is not better with PCA, the maximum F1-score is 0.835 with four PC's used. For the SVM, performance is better with PCA, with a maximum F1-score of 0.867 with 15 components used. For the logistic regression, the highest F1-score is 0.854, which is equal to the performance without PCA.

What is notable, is the varying performance of the random forest. This is probably due to the nature of a random forest that is build from different sets of decision trees that show divided outcomes. The SVM on the other hand shows a very clear image. The model clearly benefits from the transformation of the data to Principal Components. This is probably due to two things, on one side, the data is scaled, which makes it easier to group the observation, and on the other side, by removing the variance, it becomes easier to linearly separate the data. For the logistic regression there is not much to say

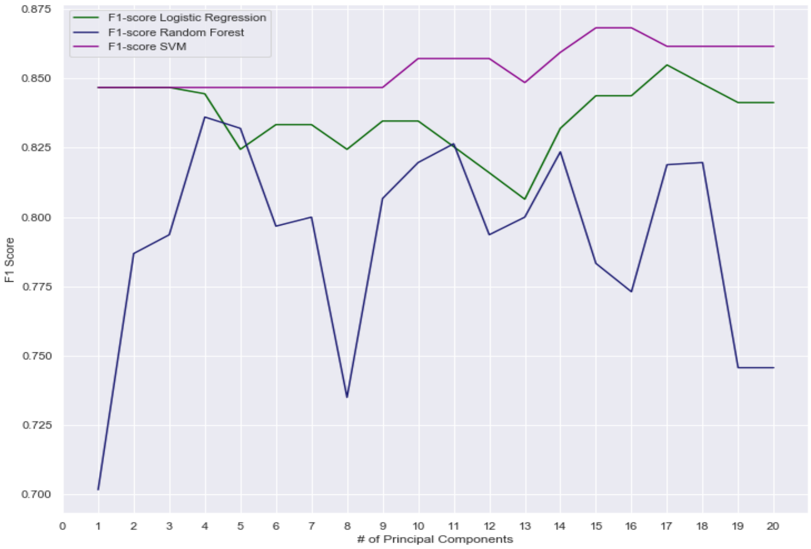

Figure 5
F1-score of different sets of PC's

other than that the performance did not improve, the table and figure do not show any other insights.

Because the risk of imbalance is also the case for the classification with PC's, a Confusion Matrix is added to the appendix to see how the classifiers predicted the observations. Table 10 in the appendix shows the Confusion Matrices with the same sets of components as Table 6. The confusion matrices shows that although the F1-score was better than the baseline, the logistic regression and the SVM predicted all observations as the majority class when using 3 PC's. As there are more PC's added, the balance gradually returns slightly in the predictions, although still heavily in favour of the majority class.

This is an indication that the models might perform better when adding the weights that were found in the previous section on class balancing. Figure 6 shows the F1-score of the balanced models. The graph shows indeed that there is an improvement of the F1-score when class weights are adjusted. The SVM has the highest F1-score of 0.872 when 13 components are used, the random forest achieved a highest F1-score of 0.854 with 7 components used and the logistic regression has a highest F1-score of 0.852 which is a slight decrease of performance with 11 components used. These results correspond with the results that were found in the previous section on class weights where the logistic regression did not benefit either from the adjustment of class weights.

One thing that is notable is that the adjustment of class weight did lead to a higher performance with less components used. This means that correctly using class weights makes dimensionality reduction possible while increasing or maintaining performance of the models. Although the performance of the SVM increased by transforming the data

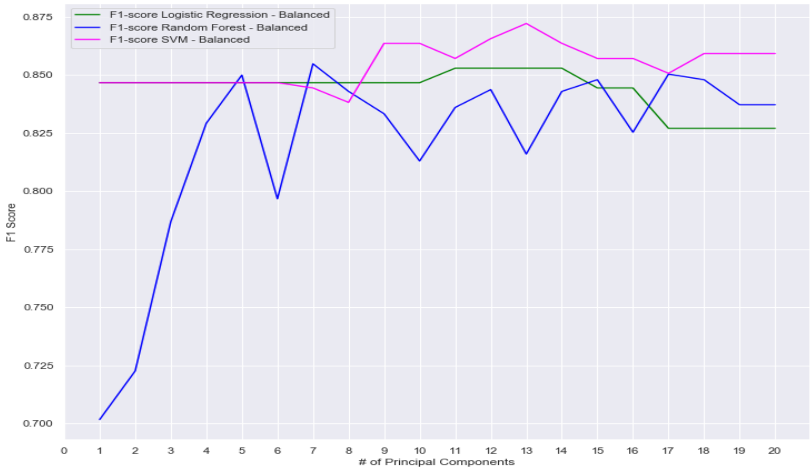

Figure 6
F1-score of different sets of PC's with balanced models

into principal components, the random forest with adjusted class weights still achieved a higher F1-score of 0.880.

5.4 Feature Importance

As the random forest is the best performing model both with and without class balancing, this model is used to calculate the feature importance. As the random forest is trained on entropy, the feature importance is calculated with Information Gain (IG).

Figure 7 shows the Information Gain per feature. From the figure can be concluded that Zone1-1week is by far the feature with the highest IG. Other features that have a relatively high IG are Zone1-1week, Zone1-8week, Zone1-1week, sRPE-4week, Zone2-4week and sRPE-8week. When looking at the density plots of the features in the appendix (9), the density plot show confirming results. When compared with the other plots, the plots of the features with the highest IG have distributions that differ from each other while the others show a more similar distribution.

To be able to see if the selected features differ in their mean value per output class, Table 7 shows the information gain and mean values of both output classes for the important features. From this table we can conclude that for all zone 1 variables, the mean value of the good performances is higher than the mean value of the bad performances. That may indicate that more training in zone 1 is beneficial for the performance but more research is needed. These insights give some guidance in what athletes and their coaches could focus on in preparation for an important race.

6. Discussion

The goal of this research was to see to what extend it was possible to predict cycling performance on the basis of training variables. This is done by looking at different

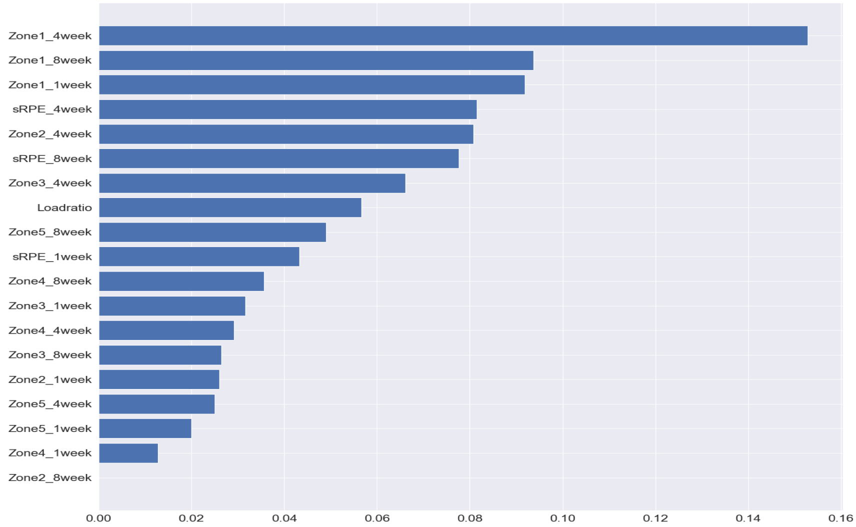

Figure 7
Information Gain per Feature

	F1-score		
Feature	**IG**	**In top-10 (1)**	**Out of top 10 (0)**
Zone1-4week	0.153	2050.0	1927.6
Zone1-8week	0.093	3907.2	3889.3
Zone1-1week	0.091	504.0	464.0
sRPE-4week	0.081	49767.7	45891.9
Zone2-4week	0.080	841.5	778.8

Table 7
Information Gain and Mean Value of top-5 features

sub-questions that each help to answer the main question. In this section, the research questions are answered and the findings in this study are placed in context with the existing literature.

6.1 Research Question 1

The first research question states: *Which model is best suited when evaluated by the F1-score?*. The models that are tested are a logistic regression, SVM and random forest. When looking at the results, the F1-scores are 0.84, 0.79 and 0.85 respectively for the models without balance. All models did perform better than the baseline of 0.734. Based on these results, the random forest is best suited to predict race performance.

Especially because the results showed that the SVM and logistic regression showed a clear preference towards the majority class.

6.2 Research Question 2

The second question states: *What is the effect of balancing the classes and which method is best suited for that? The methods that are tested are weight adjustment and SMOTE*. Overall, the adjustment of class weights shows an increase of performance of all models with the random forest again achieving the best F1-score of 0.88. Although this method showed an increase in performance for all models, the confusion matrix showed that the class weights of the logistic regression and SVM led to even more emphasis on the majority class. This means that this approach works counterproductive for these models.

The SMOTE led to a decrease in performance with only the random forest still performing better than the baseline. This is surprising as the study of (Picek et al. 2019) showed that SMOTE outperformed the class weights. One of the reasons why this is not the case for this dataset is that the size is too small.

To answer RQ2: the effect of the balancing methods is mixed; the class weights lead to an increase in performance but only laid more emphasis on the minority class for the random forest and the SMOTE leads to a decrease in performance.

6.3 Research Question 3

The third research question states: *To what extend does Principal Component Analysis contribute to the performance of the models?*. The results showed that PCA had varying effects on the different models. For the random forest, the PCA was not beneficial while the performance of the SVM increased by the PCA. When looking back at the literature from the Related Works section, this finding is not surprising. In the article (Reddy et al. 2020) already stated that PCA was less helpful for a random forest. They also stated that the classifiers with PCA performed better than the classifiers without PCA. This thesis does not confirm these results but that could be due to the fact that, again, the size of the dataset was limited as they also found that when the size is too small, PCA had a negative impact. In that way, the outcomes of that study are confirmed.

The results also showed that class balance helped with the PCA and especially made it possible to maintain a high F1-score while using less PC's. To give a concluding answer on this question: the PCA is beneficial for the SVM while it did not increase the performance of the logistic regression and the random forest.

6.4 Research Question 4

The fourth research question states: *Which features are important in the prediction of performance?*. The features that were found as important were; Zone1-4week, Zone1-8week, Zone1-1week, sRPE-4week, Zone2-4week and Zone2-4week. The findings show that it is important to have a good base in zone 1 as zone 1 of all weeks is selected.

In the Related Works section, the article of Dijkhuis et al. discussed the features that were important in the prediction of running injury based on Training Load. Based on these findings, Loadratio was also included in this dataset. Table 7 shows that Loadratio has an information gain of 0.05, which is quite low. This finding did not correspond with the findings of Dijkhuis et al as well as the other features that were selected in that paper. That could be because injury prediction is different from performance prediction or because the datasets are simply different.

6.5 Main Question

The first three sub-questions each explored ways to find the best method to answer the main question: *To what extent can race performance be predicted in cycling, based on training load?*. The eventual answer is: to quite some extend. The random forest with class weight balancing gave the best F1-score of 0.88, which is well above the baseline. The other models performed above the baseline as well but seemed to keep their preference towards the majority class, which makes the random forest better suited to the situation.

6.6 Implications, Limitations and Future Research

The results of this study are quite significant when compared with the performances of the studies of Dijkhuis et al. and Orie et al.. This shows that it is to some extend possible to make predictions about race performance based on Training Load, which would be very helpful for professional cycling team as they could base their team setup on these results.

However, this study only looked at features related to Training Load. As stated in the research of (Philips and Hopkins 2020), Training Load and the features are intertwined with other aspects that impact race performance such as team work. A recommendation for future research is to include more external variables or to standardize a performance protocol. This makes a separation between the real-life setting with real-life results and a lab-study where the isolated effect of only training variables can be measured.

Another limitation of this study was the size of the dataset. As this data is very specific for one person, the predictions will be specific as well. This makes it hard to generalize the results and say something about performance in general based on TL and eventually make a training schedule based on these findings. A recommendation for future research is to collect more data of this cyclist as well as data of other female cyclists to make the results more generalizable. Another recommendation that would be helpful for riders and their coaches is to do more research about the importance of the features. This study has given some indication of what features are important for the prediction of injuries, but research in how to optimize these parameters would probably also optimize performance.

7. Conclusion

This research has shown different techniques to make predictions on a small dataset in order to predict cycling performance in races, based on training variables. In order to do that, a logistic regression, support vector machine and random forest are trained. Secondly, two balancing methods were tested; an 'in-model' class weight adjustment and SMOTE. Thirdly, PCA was performed to see if dimensionality reduction led to an increase of performance. Lastly, the important features were identified to help athletes and coaches in optimizing training schedules.

The baseline for this research was the majority class, which was 0.734 and the models were evaluated by the F1-score with leave-one-out cross-validation. The best performing model was the random forest with class weight adjustment with an F1-score of 0.880. Overall, the models did not benefit from SMOTE and only the random forest benefited from the class weight adjustment. When the data was transformed to PC's, only the SVM showed an increase in performance with an F1-score of 0.872. The features that were found important were zone 1 of all weeks, sRPE-4week and Zone2-4week. It would be interesting to see if these results still hold up when there is more data available so that the results become more generalizable.

The results of this study can be used as a guide for classification of small datasets. For the sport-related field the implications of the results are not significant but they show that the set of features selected by the random forest might indicate that they can be important for predicting performance. Future research could show what the optimal value for these variables is in order to be classified as a top-10 result. These values can then be used as a goal to plan a training schedule around.

References

Aldehim, Ghadah and Wenjia Wang. 2017. Determining appropriate approaches for using data in feature selection. *International Journal of Machine Learning and Cybernetics*, 8.

Borg, Gunnar. 1998. *Borg's perceived exertion and pain scales.* Human kinetics.

Calvert, Thomas W, Eric W Banister, Margaret V Savage, and Tim Bach. 1976. A systems model of the effects of training on physical performance. *IEEE Transactions on systems, man, and cybernetics*, (2):94–102.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Dijkhuis, Talko, Ruby Otter, Hugo Velthuijsen, and Koen Lemmink. 2017. Prediction of running injuries from training load: a machine learning approach. *Hanze University of Applied Sciences, Institute of Sport Studies*.

van Erp, Teun. 2020. Load, intensity and performance in professional road cycling.

Khoshgoftaar, Taghi M, Kehan Gao, and Naeem Seliya. 2010. Attribute selection and imbalanced data: Problems in software defect prediction. 1:137–144.

Lever, Jake, Martin Krzywinski, and Naomi Altman. 2016. Points of significance: model selection and overfitting.

Meyer, David and FH Technikum Wien. 2015. Support vector machines. *The Interface to libsvm in package e1071*, 28.

Nagpal, Anuja. 2020. L1 and l2 regularization method.

Orie, Jac, Nico Hofman, Laurentius A Meerhoff, and Arno Knobbe. 2020. Training distribution in 1500-m speed skating: A case study of an olympic gold medalist. *International Journal of Sports Physiology and Performance*, 1(aop):1–5.

Philips and Hopkins. 2020. Determinants of Cycling Performance: a Review of the Dimensions and Features Regulating Performance in Elite Cycling. *Sports Medicine*, 6(23).

Picek, Stjepan, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. 2019. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):1–29.

Reddy, G Thippa, M Praveen Kumar Reddy, Kuruva Lakshmanna, Rajesh Kaluri, Dharmendra Singh Rajput, Gautam Srivastava, and Thar Baker. 2020. Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8:54776–54788.

Sanders D, Hesselink MK Myers T Akubat I, Abt G. 2017. Methods of monitoring training load and their relationships to changes in fitness and performance in competitive road cycling. *Int J Sports Physiol Perform*, 1(23).

Shaikhina, Torgyn, Dave Lowe, Sunil Daga, David Briggs, Robert Higgins, and Natasha Khovanova. 2015. Machine learning for predictive modelling based on small data in biomedical engineering. *School of Engineering, University of Warwick*, 48(20).

sklearn. 2020. sklearn.svm.svc.

Stoeber, Joachim, Mark A Uphill, and Sarah Hotham. 2009. Predicting race performance in triathlon: The role of perfectionism, achievement goals, and personal goal setting. *Journal of Sport and Exercise Psychology*, 31(2):211–245.

Thanh Noi, Phan and Martin Kappas. 2018. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*, 18(1):18.

White, Ryan. 2020. Acute:chronic workload ratio.

Wold, Svante, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Yildirim, Soner. 2020. How is logistic regression used as a classification algorithm?

Yiu, Tony. 2020. Understanding random forest.

Zhu, Min, Jing Xia, Xiaoqing Jin, Molei Yan, Guolong Cai, Jing Yan, and Gangmin Ning. 2018. Class weights random forest algorithm for processing class imbalanced medical data. *IEEE Access*, 6:4641–4652.

Appendix

Table 8
Original Variables, sRPE = session Rate of Perceived Exertion

Nr.	Name	Type	Description
1	Indexnr	Int	Days from the starting point. 0 = 01-11-2011
2	sRPE-1week	float	Sum of sRPE of all training sessions of one week prior to the race.
3	sRPE-4week	float	Sum of sRPE of all training of four weeks prior to the race.
4	sRPE-8week	float	Sum of sRPE of all training of eight weeks prior to the race.
5	Zone1-1week	float	Sum of minutes ridden in zone 1 intensity one week prior to the race.
6	Zone2-1week	float	Sum of minutes ridden in zone 2 intensity one week prior to the race.
7	Zone3-1week	float	Sum of minutes ridden in zone 3 intensity one week prior to the race.
8	Zone4-1week	float	Sum of minutes ridden in zone 4 intensity one week prior to the race.
9	Zone5-1week	float	Sum of minutes ridden in zone 5 intensity one week prior to the race.
10-14	Zone..-4week	float	Same as one week variable but then sum of 28 days.
15-19	Zone..-8week	float	Same as one week variable but then sum of 56 days.
20	Adj. perf.	float	Performance adjusted by Race variables.
21	Dist-tolast30	float	Distance in km until the last 30 min. of the race.
22	Dist-inlast30	float	Distance in km in last 30 min. of the race.
23	KJtolast30	float	KJ spent until last 30 min. of the race.
24	KJinlast30	float	KJ spent in last 30 min. of the race.
25	HMtolast30	float	Altimeters until the last 30 min. of the race.
26	HMinlast30	float	Altimeters in the last 30 min. of the race.
27	KJperKMtolast30	float	KJ spent per km until last 30 min. of the race.
28	KJperKMinlast30	float	KJ spent per km in last 30 min. of the race.

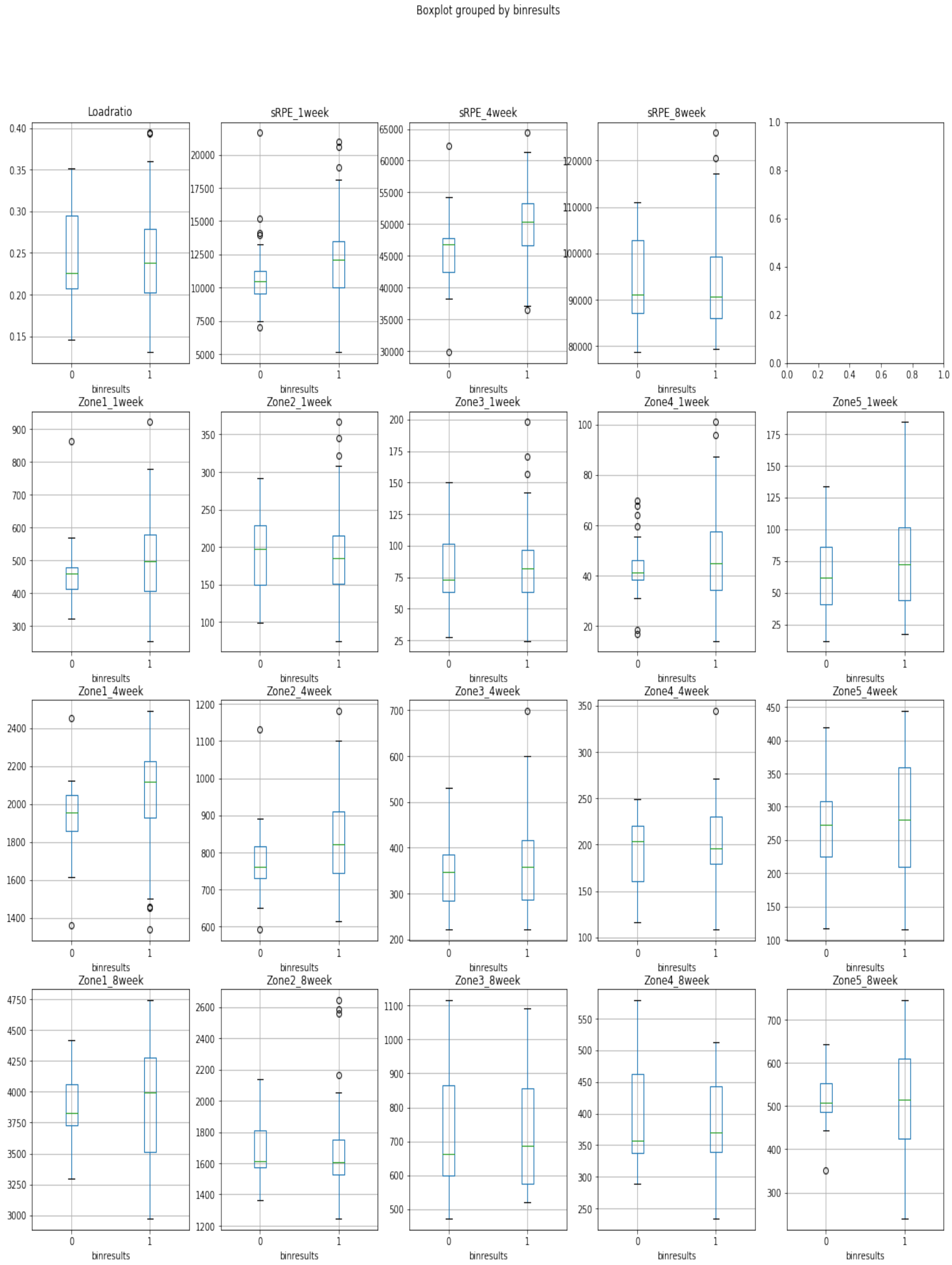

Figure 8
Boxplots per output class before outlier removal

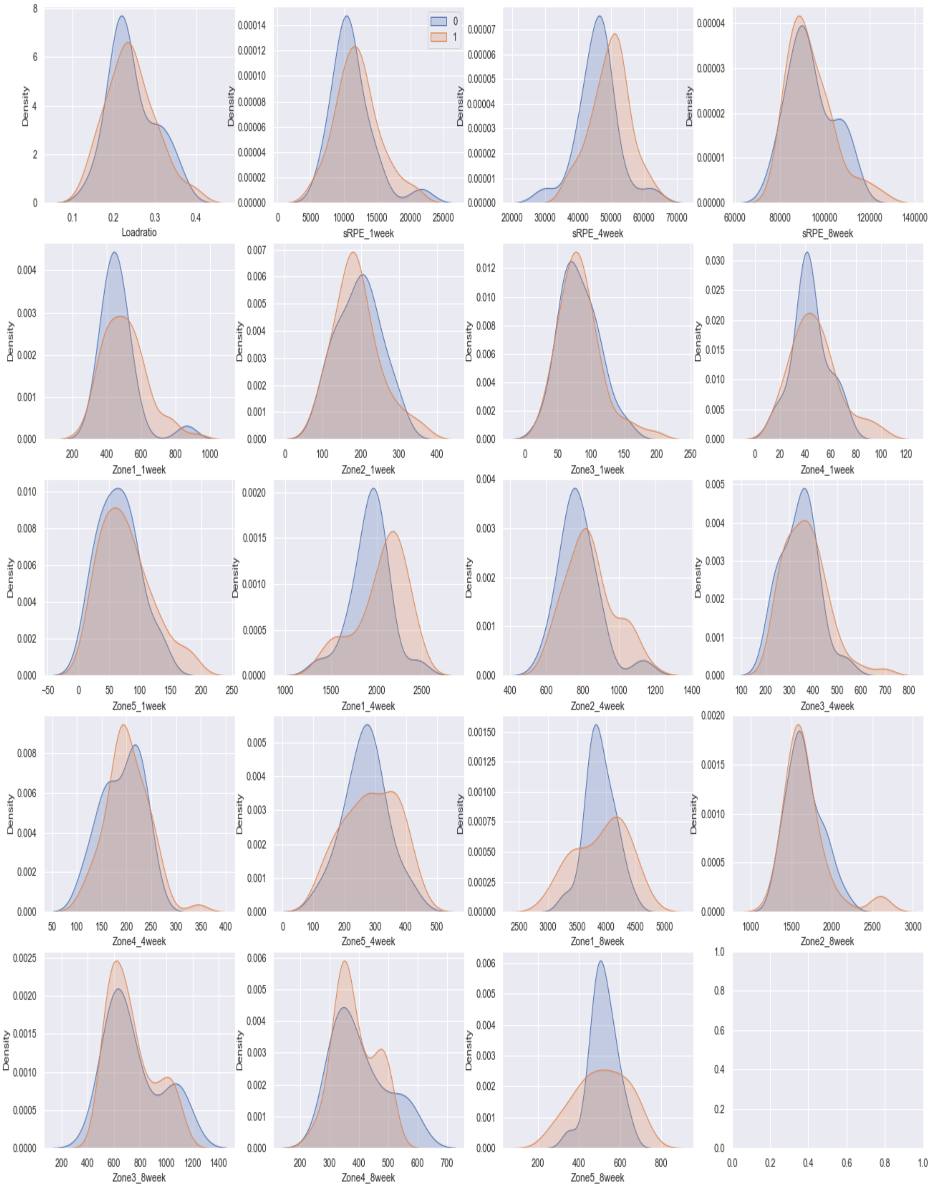

Figure 9
Density plots of all features seperated by binresults

Table 9
Confusion Matrices of the Models without Feature Selection

Simple models		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	52	14	0.79	Predicted	1	46	12	0.79	Predicted	1	52	13	0.80
	0	6	7	0.53		0	12	9	0.43		0	6	8	0.57
Recall		0.90	0.33		Recall		0.79	0.43		Recall		0.90	0.38	
Class weights		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	57	20	0.74	Predicted	1	51	12	0.81	Predicted	1	53	10	0.84
	0	1	1	0.50		0	7	9	0.56		0	5	11	0.69
Recall		0.98	0.48		Recall		0.88	0.43		Recall		0.91	0.52	
SMOTE		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	38	16	0.70	Predicted	1	29	13	0.68	Predicted	1	42	6	0.88
	0	20	42	0.68		0	29	45	0.61		0	16	52	0.76
Recall		0.66	0.72		Recall		0.50	0.76		Recall		0.72	0.90	

Table 10
Confusion Matrices of the Models with PCA

3 Components		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	58	21	0.73	Predicted	1	58	21	0.73	Predicted	1	50	18	0.74
	0	0	0	0.0		0	0	0	0		0	8	3	0.273
Recall		1	0		Recall		1	0		Recall		0.862	0.143	
6 Components		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	55	19	0.74	Predicted	1	58	21	0.73	Predicted	1	49	16	0.75
	0	3	2	0.40		0	0	0	0		0	9	5	0.36
Recall		0.95	0.09		Recall		0	0		Recall		0.85	0.24	
10 Components		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	53	16	0.77	Predicted	1	57	18	0.76	Predicted	1	50	14	0.78
	0	5	5	0.5		0	1	3	0.75		0	8	7	0.47
Recall		0.91	0.24		Recall		0.98	0.14		Recall		0.86	0.33	
16 Components		Actual					Actual					Actual		
Log. Reg.		1	0	Precision	SVM		1	0	Precision	Random Forest		1	0	Precision
Predicted	1	54	16	0.77	Predicted	1	56	15	0.79	Predicted	1	46	15	0.79
	0	4	5	0.56		0	2	6	0.75		0	12	6	0.33
Recall		0.93	0.24		Recall		0.97	0.29		Recall		0.79	0.29	