Universiteit Leiden
The Netherlands

Opleiding Informatica

Predicting the characteristics of successful cyclists in multi-week races

Lucas van Rooij

Supervisors:
A.J. Knobbe & A.-W. de Leeuw & S. van der Zwaard

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

04/08/2019

Abstract

Due to technological advances it is possible to collect, store, and process more and more data nowadays. By applying data mining techniques to this data we can find previously unknown, and useful insights such as consumer patterns, flu epidemic predictions, or which cyclist will win a race. In this thesis, our Data Science challenge is finding out what makes a professional cyclist successful during the three main multi-week cycling races, which are also known as Grand Tours. Specifically we will use data mining techniques in order to find an answer to the question: what are the characteristics, of successful cyclists in a Grand Tour? To find these characteristics we have started by scraping cycling performance statistics from a website, and from this raw data we have built features that are related to a cyclist's physique, form, and experience. Since our data set contained a class imbalance, we first applied SMOTE to obtain a data set with a similar number of successful and unsuccessful cyclists. Hereafter we have built a decision tree that we have used to gain insight in which characteristics determine whether a cyclist is in the top-10 or not.

After comparing models with different splitting criterion, and and class balancing techniques we obtain one final model. This model has an AUC-score of 0.905. From our final model we have concluded that the most important characteristic upon deciding the successfulness of a cyclist is that he has obtained a top-10 finish in the general classification in a Grand Tour in the past two years. This implies that strong performances in the past two years determine whether a cyclist will be successful nowadays. When leaving previously achieved results out of our data set we obtain a model that has an AUC-score of 0.77. This model shows us that cyclist's that have a BMI value ≤ 21.16, and have participated in at least 3 Grand Tours have a large chance of being classified as top-10.

Contents

1 **1**

1.1 Introduction . . . 1
1.2 Research question . . . 2
1.3 Thesis overview . . . 2

2 Methods **4**

2.1 Decision tree . . . 4
2.1.1 Pruning . . . 4
2.2 Hyperparameter tuning: grid search . . . 5
2.3 Cross-validation . . . 5
2.4 Model evaluation . . . 6
2.4.1 Precision . . . 6
2.4.2 Accuracy . . . 7
2.4.3 Recall . . . 7
2.4.4 F1-score . . . 7
2.4.5 False positive rate . . . 7
2.4.6 Confusion matrix . . . 7
2.4.7 ROC-curves and AUC . . . 8
2.5 Influence of an unbalanced data set . . . 8
2.5.1 Class-weight-based approaches . . . 9
2.5.2 Sampling-based approaches . . . 9
2.6 Influence of different splitting criteria . . . 9
2.7 Implementation . . . 10

3 Data **11**

3.1 Data collection method . . . 11
3.2 Data explanation . . . 11
3.2.1 Raw data . . . 12
3.2.2 Constructed features . . . 12
3.3 Missing values . . . 13

4 Experiments **16**

4.1 Decision trees with complete data set . . . 16
4.1.1 Decision tree quality metrics . . . 16
4.1.2 Visualising the decision tree . . . 19

4.2 Decision trees with modified data set 20
4.2.1 Quality metrics 21
4.2.2 Visualising the decision tree 23

5 Discussion **25**

6 Conclusion **27**
6.1 Future work 27
References 29

Chapter 1:

1.1 Introduction

The importance of data, and data mining has been growing rapidly over the past years. In both business and academics, data mining has gained huge potential due to technological advances. This growth has been facilitated by the improvement of data collection devices such as sensors and trackers. The ability to store more and more data than ever before is also part of this growth. Where 1 GB of storage used to cost $1.000.000 back in 1980, 1 GB of storage now only costs $0,10. The huge amounts of data that are collected and stored nowadays are also processed a lot faster. The amount of transistors in a 1980 CPU used to be around 50, and has been increasing exponentially up to 10.000.000 transistors nowadays, making data processing much faster. The last component that has been growing rapidly over the past years is the amount of data visualisation tools. As more and more data visualisation tools such as MATLAB, and Sisense are available on the market there is more potential to maximise the use of data collection, data storage, and data processing [Luo08]. These developments have lead to a growth in applications of data mining in almost any field ranging from finance [GG05], to medical healthcare [DR13], and sports [McC10].

As sports are gaining popularity all over the world, many organisations are willing to invest funds in order to achieve better match results. For different sports organisations, this has caused an increased interest in predicting the results of sports matches. These predictions can be done using data mining techniques such as decision trees, logistic regression, and Bayesian analysis [Ras13]. Besides that these techniques can predict the results of sports matches [RB19], these techniques can also help us find useful insights such as information about optimal training schedules [Roz17], or optimal game tactics [Ofo13]. Data mining techniques are also being used to find out what the characteristics of successful athletes are. This has for instance been done for American football players [Spi07], judoka's [Hoa91], and runners [TN89], but has not yet been done for every sport.

Knowing what the characteristics of successful athletes are, is important in any field of sports, but is especially interesting in endurance sports such as cycling. As cycling is a sport that demands very high energy consumption over long periods of time, small changes in for example the weight of a cyclist can have a large impact on his performance during a race [Jeu01]. Not just the weight of a cyclist influences his performance in road races. Previous studies have shown that the whole physical profile of a cyclist determines how well a cyclist will perform [nMP01]. This physical profile of a cyclist is often related to their speciality. Cyclists that excel at sprinting for example, tend to have a higher absolute maximal power output (W_{max}), while cyclists that excel at climbing have a higher maximal power output per kilogram of body weight (W_{max}/kg) [SP99].

How well a cyclist performs is not only defined by his physical characteristics. As experience and form play a large role in professional achievement [Eri07], we expect this to also influence the performance of professional cyclists. Current research does not yet tell us how future performance can be predicted from experience in the preceding years, and form in the current year. Experience and form can come in different appearances for a cyclist. Experience for instance, can be the number of years that a rider has been cycling as a professional. An example of the form of a rider is the number of races that a cyclist has won this year. The challenge that now lies ahead of us is finding out what experience, and what form contribute to a cyclist's success.

In this thesis, we will focus on cyclists that try to finish as high as possible in the general classification as this is the main goal of a cycling race. Most cyclists that try to win a cycling race will at least try to finish in the top-10 of the final general classification. We thus define being successful as acquiring a top-10 position in the final general classification. The focus in this thesis wil lie on the three main multiple day races, as being successful in one of these races is the highest achievement a cyclist can get. These three multiple day races, known as the Grand Tours, are: La Giro d'Italia, Le Tour de France, and La Vuelta a España. In order to investigate what characteristics influence the successfulness of a cyclist we will be applying data mining techniques. Specifically, we will collect a data set that contains features about a cyclist's physique, experience, and form. From this data set we will then build a decision tree in order to predict whether a cyclist will be successful or not. When we have have highlighted the most important characteristics from the best predicting model, we can derive what the most important characteristics of a successful cyclist are.

1.2 Research question

During this thesis we are going to investigate what characteristics have influence on the successfulness of a cyclist. We thus define the research question of this thesis as: *What are the characteristics of successful cyclists in Grand Tours?* We will try to find these characteristics using a decision tree. As a decision tree can be built with different splitting criteria, and our data set contains class imbalance, we add the following sub questions.

- How does class imbalance influence the performance of our model? And what can we do to overcome this?
- What is the influence of splitting criteria on the performance of our model?

1.3 Thesis overview

We have given this thesis the following order. In chapter 2 we explain why we use a decision tree, how we optimize hyperparameters, how cross-validation is used to validate our model, how we can overcome our class imbalance problem, which splitting criteria we use, and finally what performance metrics we have used for our model. Next comes chapter 3 where we include an explanation of the data set, how the data is collected, and how we handled missing values in the data set. Hereafter

we describe the experiments and their outcome in chapter 4, followed by chapter 5 where we discuss the results from chapter 4. Finally we give a conclusion, and give possibilities for future research in chapter 6.

Chapter 2: Methods

Now that we know the goal of this thesis we have to define methods in order to reach this goal. This section will thus contain the methods that we will use in order to answer our research question, and sub questions.

2.1 Decision tree

In order to discover what characteristics define a successful cyclist we will collect a data set containing multiple features about a cyclists physique, experience and form. A more in-depth explanation of how we collect this data set, and what it contains is described in chapter 3. From this data set we have to build a model that predicts the successfulness of cyclists, and show how accurate these predictions are. From this model we can then learn what characteristics are important upon deciding the successfulness of cyclists. We thus need a model that can predict whether a cyclist will finish in the top-10 general classification of a Grand Tour or not, and also show us which decisions have been made in order to make this prediction. A decision tree is a model that fulfills this need, as decision trees are used to make predictions, and can be used to visualise which decisions have been made in order to make a prediction. We will thus use a decision tree in order to answer our research question.

As our target attribute is binary, and thus discrete, we will build a classification tree. A classification tree can be seen as a flow-chart. The classification tree starts at the root with the base test. If the test is positive, we move left in the tree, if the test is negative we move right in the tree. The nodes after the root node are called internal nodes and also contain a test on one of the features. Same concept holds here, left for a positive test, right for a negative test. Each internal node has edges to child nodes, or is a final node. The final nodes of a tree that do not grow any further are called leaf nodes, or leaves. A leaf node gives a value for examples that end up in the leaf node when following a path started in the root node. A decision tree can be built by splitting the original data set into smaller subsets by performing a value test on one of the features. How the model is made is based on splitting the data, with the most important splits on the top of the tree. Which splits are the most important is determined by splitting criteria. More on splitting criteria in section 2.6.

2.1.1 Pruning

Recursively expanding the tree can quickly result in a large and unclear decision tree overfitted to the part of the data set that is used for training the model. In order to prevent this, we have to

prune our tree. Pruning can be done after the tree is built, this is called post-pruning. Post-pruning evaluates the tree after it is completely built, and then removes insignificant branches. Branches can be insignificant because they have a very low amount of samples, or because they have very low information or Gini gain. Pruning can also be done during the construction of the tree, this is called pre-pruning. Pre-pruning prevents the tree from generating insignificant branches by setting stopping criteria. Stopping criteria can be a maximum depth of the tree, or a minimum amount of samples that need to be in a child node. When one of these criteria is met, the tree stops expanding a certain path and creates a leaf node.

2.2 Hyperparameter tuning: grid search

In order to find what parameters belong to the best possible model, we set up a grid search that simply tries every single combination of parameters. The maximum depth of a tree, or `max_depth`, is one of the parameters that we will optimize. Furthermore we optimize `min_samples_split` which is the minimum amount of samples required to split an internal node. When we use class weights as balancing technique we also optimise the weights for both the no top-10 class, as well as the top-10 class.

2.3 Cross-validation

When building our model we split our data set into a training, and test set. The training set is used to build our model, and the test set is used to measure the performance of the model on unseen data. The usual split ratio is 90% for training, and 10% for testing. This approach tests the model on only one part of the data set. This test set can contain a specific subset of the complete data set that might not be representative for the complete data set. Using only one part of the data set can possibly give a poor impression of how well the model will perform on new unseen data. To overcome this problem we use k-fold cross-validation [Koh95]. This method does not do one, but more folds into training and test data. This can be any k number of splits, but the most common number for k is 10. We divide our training set into k parts, and then build k different models. Each time a different $\frac{9}{10}$th of the training set is used as training fold, and a different $\frac{1}{10}$th is used as test fold. We validate our model k times, and get the average estimate of our model performance. This way we use our complete data set for building our model.

Figure 2.1: 10 fold cross-validation iterates 10 times. Each iteration a different $\frac{1}{10}$th of the training set is used a test set. This way the complete training set is used for evaluating a model. [Ros16]

2.4 Model evaluation

Now that we know how to build a decision tree we have to define measures in order to determine how well our model performs. The model can predict a sample in four possible ways.

- True positive (TP): a sample that is positive (top-10), and is classified correctly as positive.
- True negative (TN): a sample that is negative (no-top 10), and is classified correctly as negative.
- False positive (FP): a sample that is negative, and is classified incorrectly as positive.
- False negative (FN): a sample that is positive, and is classified incorrectly as negative.

Of course we want our TP and TN to be high, and our FP and FN low. But these measures are absolute, and can be biased by the distribution of samples in our test set. This is why we introduce relative measures.

2.4.1 Precision

We use the precision of a model in order to find the model's capability of not qualifying a sample as positive that is negative. Precision can be calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

It's value lies between 0 and 1 with 0 being the worst possible outcome, and 1 being the best.

2.4.2 Accuracy

The accuracy is a model's capability to correctly classify a sample, and is calculated as follows:

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

We will not use this metric, as our test data set contain relatively many negative samples. Causing the accuracy to say more about correctly classifying negative samples, than correctly classifying both samples.

2.4.3 Recall

The recall, or True Positive Rate (TPR), of a model is the model's capability to find all positive samples, and can be calculated as follows:

$$Recall = \frac{TP}{TP + FN}$$

It's value lies between 0 and 1 with 0 being the worst possible outcome, and 1 being the best. We use this metric as we want to know how good our model is at finding all positive (top-10) cases.

2.4.4 F1-score

The F1-score of a model is the weighted average of precision combined with recall. F1-score can be calculated as follows:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

It's value lies between 0 and 1 with 0 being the worst possible outcome, and 1 being the best. We use f1-score as scoring method for each fold in our cross-validation.

2.4.5 False positive rate

The False Positive Rate (FPR) is the ratio between the number of negative samples classified as positive divided by the total number of actual negative samples, and can be calculated as follows:

$$FPR = \frac{FP}{FP + TN}$$

2.4.6 Confusion matrix

These model evaluation parameters are summarized in a confusion matrix to show the performance of our model on the test set.

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

Table 2.1: The layout of a confusion matrix. The columns of the table represent the predicted class, while the rows of the table represent the actual class.

2.4.7 ROC-curves and AUC

In order to choose the best model as our final model we will use the Receiver Operating Characteristic (ROC) curve and it's corresponding Area Under the Curve. The ROC curve is created by plotting the True Positive Rate (TPR) on the y axis against the False Positive Rate (FPR) on the x axis. The curve shows the performance of our model at different parameter settings, such as splitting criteria, or class balancing techniques. The AUC is the entire area under the ROC curve and lies between 0 and 1. An AUC value of 0 represent the worst possible model, while an AUC value of 1 represents the best possible model.

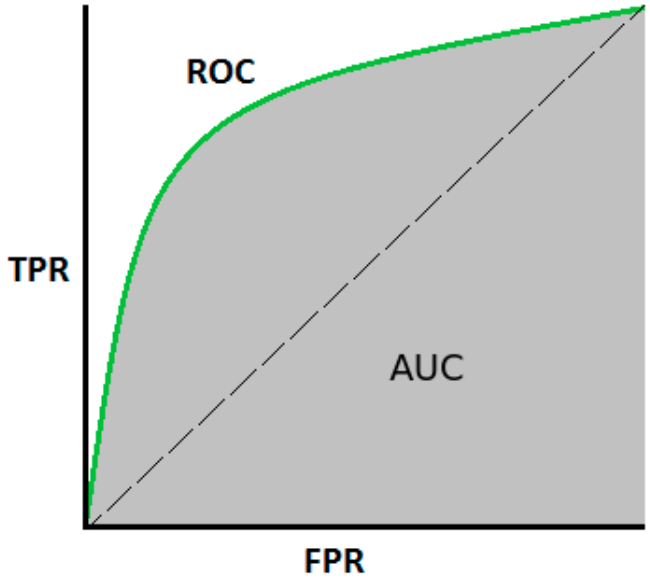


Figure 2.2: ROC curve and AUC [Nar19].

2.5 Influence of an unbalanced data set

One of the most common issues for classification problems is an imbalance in the data set. An imbalance in the data set means that one class is more represented than the other. If a data set is balanced, this means that each class accounts for 50% of the data points. A small imbalance, for example a 60%/40% ratio would not cause large problems, but when the ratio is 90%/10% classification trees will have trouble learning the minority class. Most of the times the majority class is the negative case, and the minority class is the positive case. When using a traditional

cost-insensitive classifier the model will likely predict most examples as the majority class, and barely predict the minority class. In order to tackle this problem we introduce two options.

2.5.1 Class-weight-based approaches

The class-weight-based approach gives different weights to each target class. It penalizes the model heavier for making errors on the minority class. If we take an example data set that contains 3000 "false" samples and 300 "true" samples we can determine a class weight by picking a number that lies between the ratio of falses to trues. In this case a number between 1 and 10. The weight will penalize the classification function for misclassification of the minority cases.

2.5.2 Sampling-based approaches

There are three ways to use the sampling based approach:

- Oversampling; adding more samples of the minority class in order to bring balance in the data set. By oversampling we simply duplicate samples from the minority class. The downside of this is that we risk overfitting to a few number of samples.
- Undersampling; removing samples of the majority class in order to bring balance in the data set. By removing samples from the data set we risk removing representative, and thus useful information.
- Combination of both; has the advantage of both methods, and less of the disadvantages.

The most used approach nowadays to handle the problem of class imbalance is called Synthetic Minority Over-Sampling (SMOTE) [NC02]. SMOTE makes use of oversampling, but is not done by simply copying samples from the minority class. SMOTE generates new samples by taking every minority class sample and using it's k nearest neighbours. It then takes the difference between the feature vector of the minority class sample, and the feature vector of one of the k nearest neighbours. Then it multiplies this difference by a random number between 0 and 1. The new sample is then created by adding this difference to the original minority sample. The standard value for k is 5, but can be altered in most implementations of the SMOTE algorithm. The SMOTE algorithm does not simply copy samples but creates synthetic new samples that are similar to the existing samples. The synthetic examples cause the classifier to create larger and less specific decisions, and thus prevent overfitting.

2.6 Influence of different splitting criteria

There are multiple common splitting criteria used for building a decision tree.

- The first one being *information gain*. Information gain is based on the reduction of entropy after a split on a certain attribute. Entropy, also known as uncertainty, is a value for the amount of chaos or impurity in a data set, and measured in bits of information [Ren61]. Impurity defines the distribution of the target, in our case top-10 or no top-10, in a data set.

The entropy of a data set S can be calculated as follows: let p_i be the probability of attribute i in data set S then:

$$Entropy(S) = \sum_{i=1}^{n} -p_i log_2(p_i)$$

A branch with entropy 0 is a leaf node and can not be further expanded. A branch with entropy > 0 can, but is not required to, be further expanded.

Now that we know the entropy of our data set we can calculate our information gain to find out which attribute split gives us the largest decrease in impurity. Information gain is calculated by splitting the data set on different attributes. The entropy for each branch is then calculated and added in proportion to get the total entropy after a split on a certain attribute. The resulting entropy is subtracted from our base entropy before the split resulting in the amount of information gain. The higher the information gain, the better the split. If S is our data set, and X is an attribute that we split on, then the formula for information gain is:

$$InformationGain(S, X) = Entropy(S) - Entropy(S, X)$$

- A different, but also commonly used splitting criterion is Gini gain. Suppose that we randomly pick an instance from our data set, and randomly classify it according to the target distribution from our data set (10/1 for our data set). Then the probability of us classifying this instance correctly is called the Gini impurity [RS04a]. Let C be the amount of classes, and $p(i)$ the probability of picking an instance with class i then the Gini impurity G can be formulated as:

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

When calculating the Gini impurity after a split it is weighted to the amount of samples that remain after the split. We can now calculate the Gini gain by subtracting the weighted Gini impurity after the split from the Gini impurity before the split. The higher the gain, the better the split.

This process continues recursively on the subsets until a stopping criterion is reached. Example stopping criteria are, minimum samples per leaf, and maximum depth of the tree.

2.7 Implementation

Python's `sklearn.tree.DecisionTreeClassifier` package gives us all the tools that we need in order to implement a decision tree with our data set. It lets us set parameters so that we can try different splitting criteria, and multiple different stopping criteria.

Chapter 3: Data

This chapter explains how we have collected our data set, what the collected data means, and how we overcome the issue of missing values.

3.1 Data collection method

In order to find out what characteristics belong to a successful cyclist we have to obtain a data set that contains features about a cyclist's characteristics. We want the data set to include features about form, experience and physique.

Information about the experience and form of professional cyclists is widely available to the public. The website *www.procyclingstats.com* provides us with the results of all major professional cycling races that have been held over the past ≈ 100 years. However, not all historical information is relevant today as professional cycling races from a long time ago are not representative for current races. Riders nowadays use better bikes for instance, also length and intensity of the parkour has changed, training methods have improved, and even the rules have altered. That is why we have chosen to gather the results of Grand Tours in the past 10 years (2009-2018), because this time period is the most representative for current cycling.

Furthermore, the website provides some basic physiological characteristics of all cyclists such as weight, and height. The data is accessible by browsing the website, but is not yet in the correct format to pass on to a classification algorithm. In order to get this data in the correct format we build a web scraping tool. We can now get raw data from the website, and from this raw data we can then build features about physique, form, and experience. Python's BeautifulSoup package collects the HTML source code from a web page, and puts it in an easy to search format. This gives us the ability to quickly scrape data from the website, and put it in a ready to use format for a data mining algorithm.

3.2 Data explanation

Scraping the website delivered us raw data that was mostly not yet ready to be processed by a decision tree. From the raw data we were able to extract some features that were ready to use, but most of the data was not yet ready to be processed by a decision tree. We have thus also constructed our own features.

3.2.1 Raw data

From the website we were able to collect the following raw data.

- A list with the results that a cyclist has achieved in the same year as the Grand Tour. This list includes which races he has participated in, what position he acquired per this race, what the distance was of this race, and what the date was of this race.
- A list with the results that a cyclist has achieved in the year before the Grand Tour. Also containing the races he has participated in, the position he has acquired in this race, the distance of the race, and the date of the race. This list also contains the feature `Total kms last year`.
- A list with all the Grand Tours in which a cyclist has started. This list also contains `All time GT starts`.
- A list with physical information about the cyclist. This contains the features `Age`, `Weight`, and `Height`.

3.2.2 Constructed features

The following features are also included in the data set, but were not available as raw data. We have thus constructed these features ourselves, from the available raw data.

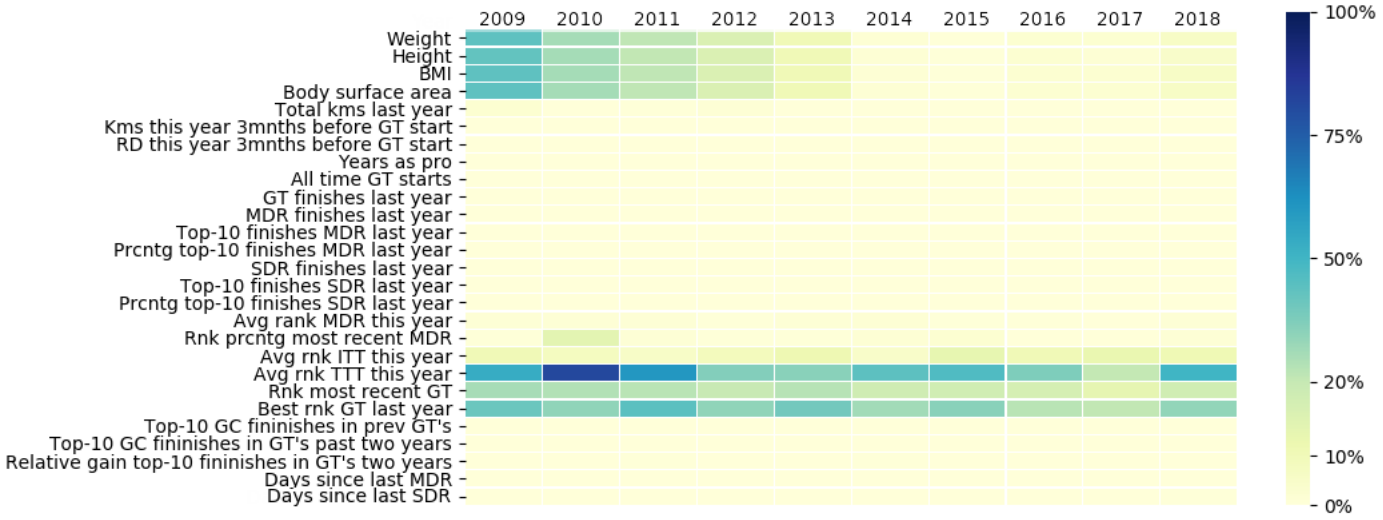
- `Rnk`:the binary target of this data set. If a rider finished in the top-10 of this grand tour, the value is 1. Otherwise it is 0.
- `BMI`:Body mass index, calculated as follows: $BMI = \frac{weight(kg)}{length^2(m)}$.
- `BSA`:Body surface area. According to Du Bois it is calculated as follows: $BSA = Weight^{0,425} * Height^{0,725} * 71,84$ [dBdB16]. This formula gives an approximation of the body surface area in square centimeters when weight is entered in kilograms, and height in centimeters. It is estimated that the formula makes a $\pm 1,5\%$ error when compared to measuring the BSA.
- `Years as pro`: the number of years that a cyclist has competed in professional cycling.
- `Kms this year in 3mnths before GT start`: the number of kilometers that a rider has cycled in professional races that took place three months before the start of the current Grand Tour (GT).
- `RD this year 3mnths before GT start`: the number of race days (RD) that a rider has had in professional races that took place three months before the start of the current Grand Tour.
- `GT finishes last year`: the number of Grand Tour finishes in the year before the current Grand Tour.
- `MDR finishes last year`: the number of multiple day race (MDR) finishes in the year before the current Grand Tour.

- `Top-10 finishes MDR last year`: the number of top-10 general classification finishes in multiple day races in the year before the current Grand Tour.
- `Prcntg top-10 finishes MDR last year`: the percentage of multiple day races in the year before the current Grand Tour in which the rider acquired a top-10 finish in the general classification.
- `SDR finishes last year`: the number of single day races (SDR) that the rider finished in the year before the current Grand Tour.
- `Top-10 finishes SDR last year`: the number of top-10 finishes in a single day race in the year before the current Grand Tour.
- `Prcntg top-10 finishes SDR last year`: the percentage of single day races in which the rider finished as top-10 in the year before the current Grand Tour.
- `Avg rnk MDR this year`: the average general classification rank in multiple day races in this year before the current Grand Tour.
- `Avg rnk ITT this year`: the average rank in individual time trial (ITT) stages or races this year before the current Grand Tour.
- `Avg rnk TTT this year`: the average rank in team time trial (TTT) stages or races this year before the current Grand Tour.
- `Top-10 GC finishes in prev GT's`: the number of top-10 general classification finishes in Grand Tours before the start of the current Grand Tour.
- `Top-10 GC finishes in GT's past two years`: the number of top-10 general classification finishes in Grand Tours in the two years before the current Grand Tour.
- `Relative gain top-10 finishes in GT's past two years`: the percentage increase in top-10 general classification finishes in Grand Tours in the two years before the current Grand Tour.
- `Days since last MDR`: the number of days since the last multiple day race before the start of the current Grand Tour.
- `Days since last SDR`: the number of days since the last single day race before the start of the current Grand Tour.

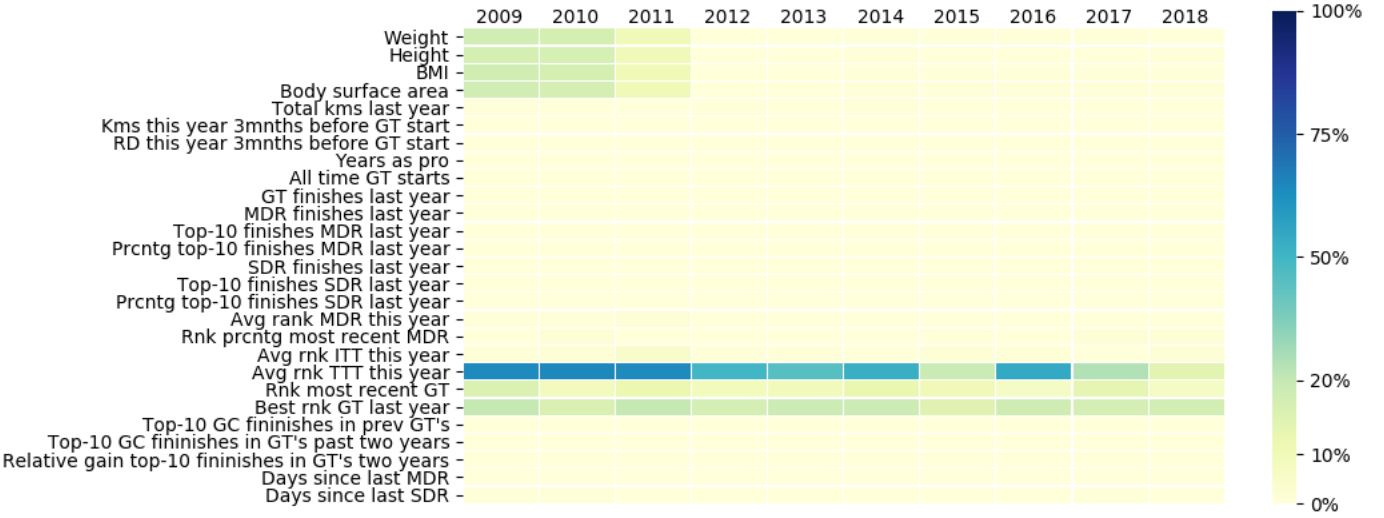
3.3 Missing values

The website used to construct this data set contains information about cyclists and professional races, but unfortunately not about every rider or every race. For some features it was not possible to collect a value for every rider in every race, simply because the value was not available on the website. This mostly happened with lesser known riders, or lesser known races. As mentioned in chapter 2 we will use this data set in order to construct a decision tree. Building a decision tree

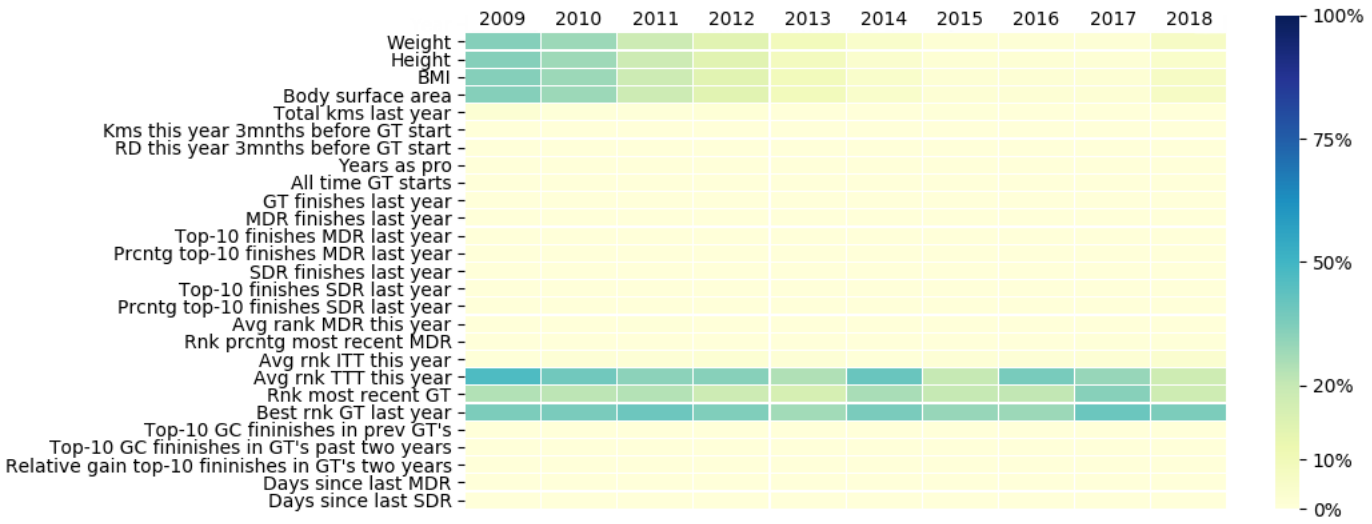
with input data that contains features with missing values requires adjustment to the algorithm, or to the data set. In order to account for missing values, we first had to make an overview of which races, which riders, and which features contained the most missing values.



(a) A plot of the missing values from La Giro d'Italia 2009-2018. 100%(blue) means all values are missing, 0% (light yellow) means no values are missing. Avg rnk TTT this year, Rnk most recent GT, and Best rnk GT last year show high missing value rates for each year. Height, weight, BMI, and BSA, have a lot of missing values only in the year 2009 up to 2012.



(b) A plot of the missing values from Le Tour de France 2009-2018. Avg rnk TTT this year, Rnk most recent GT, and Best rnk GT last year show high missing value rates for each year, but lower than figure 3.1a. Height, weight, BMI, and BSA, have a lot of missing values only in the year 2009 up to 2011.



(c) A plot of the missing values from La Vuelta a España 2009-2018. Avg rnk TTT this year, Rnk most recent GT, and Best rnk GT last year show high missing value rates for each year just as in figure 3.1a. Height, weight, BMI, and BSA, have a lot of missing values only in the year 2009 up to 2011.

Missing values can be replaced by a certain value; for instance the random, worst, or average value of a feature. Doing this, however, puts bias in the data set. In order to acquire a data set that does not contain missing values we remove the features `Avg rnk TTT this year`, `Rnk most recent GT`, and `Best rnk GT last year` from the data set as these features show high missing value rates for every year, ranging between 30% and 50%. Furthermore, we remove the years 2009, up to 2011 from the data set as these years contain high missing values rates for multiple features, namely `height`, `weight`, `BMI` and `BSA`. The resulting data set still contains some features with missing values. But the percentage of missing values for these features are relatively low ($\leq 10\%$), so completely removing these features would be a loss of potentially useful information. In order to handle the small remaining amount of missing values we exclude the remaining samples with missing data from our analysis. The final data set that now remains contains no more missing values.

Chapter 4: Experiments

In order to give an answer to our research question: "What are the characteristics of successful cyclists in Grand Tours?" we will perform multiple variants of an experiment that give answers to our sub questions mentioned in section 1.2. These sub questions are:

- How does class imbalance influence the performance of our model? And what can we do to overcome this?
- What is the influence of splitting criteria on the performance of our model?

When we have answered these sub questions, we can combine these answers in order to find an answer to our research question. For both sub questions we will show the experiments that we have performed in order to answer these questions. Hereafter we will show the results of these experiments, and finally we will visualise the best performing model in order to find the most important criterion upon deciding the successfulness of a cyclist.

4.1 Decision trees with complete data set

In order to find out how class imbalance, and class balancing techniques influence the performance of our model, we have built three decision trees. We have built one tree without applying class balancing techniques, one tree using SMOTE on the data set, and one tree by applying class weights. These three trees all have Gini gain as splitting criterion. In order to find out what the influence of splitting criteria on the performance of our model is, we have again built three decision trees. We have built one tree without applying class balancing techniques, one tree using SMOTE on the data set, and one tree by applying class weights. This time, we use information gain as splitting criterion.

4.1.1 Decision tree quality metrics

In order to quantify the performance of our models we present the tree with a test set that contains 10% of our data set, and calculate the quality metrics defined in section 2.4. We have also performed these experiments with a test set that contains 20% of our data set, this resulted in quality metris that were almost identical.

		Precision	Recall	f1-score	FPR
Gini gain	No class balancing techniques	0.64	0.31	0.42	0.02
	Class weights	0.58	0.72	0.65	0.05
	SMOTE	0.57	0.83	0.68	0.02
Information gain	No class balancing techniques	0.67	0.14	0.23	0.01
	Class weights	0.60	0.72	0.66	0.05
	Smote	0.55	0.83	0.66	0.07

Table 4.1: This table shows the quality metrics that can be calculated when the decision trees are presented with a test set that contains 10% of the total data set.

As we can see in table 4.1 there is almost no difference in performance between the decision trees that have Gini gain, or information gain as splitting criterion. However, there is a large difference in performance between the trees that have been generated with class balancing techniques, and trees that have been generated without class balancing techniques. Especially the recall, or TPR, of the trees generated with class balancing techniques is much better than the recall for the trees generated without class balancing techniques. This shows that applying class balancing techniques causes the model to perform better at classifying all top-10 samples.

Striking is, however, the fact that the precision of the trees generated without class balancing techniques, is higher than for the trees generated with class balancing techniques. This can be explained when we take a look at the confusion matrices of the trees generated without class balancing techniques.

		Predicted	
		No top-10	Top-10
Actual	No top-10	271	5
	Top-10	20	9

(a) Confusion matrix for the tree built with Gini gain, and no class balancing techniques

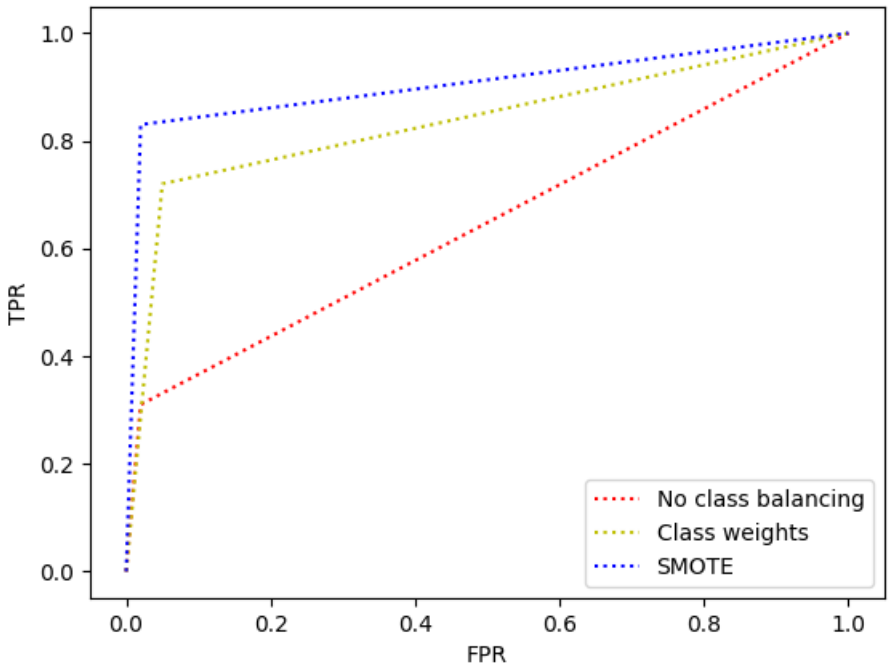
		Predicted	
		No top-10	Top-10
Actual	No top-10	274	2
	Top-10	25	4

(b) Confusion matrix for the tree in built with information gain, and no class balancing techniques

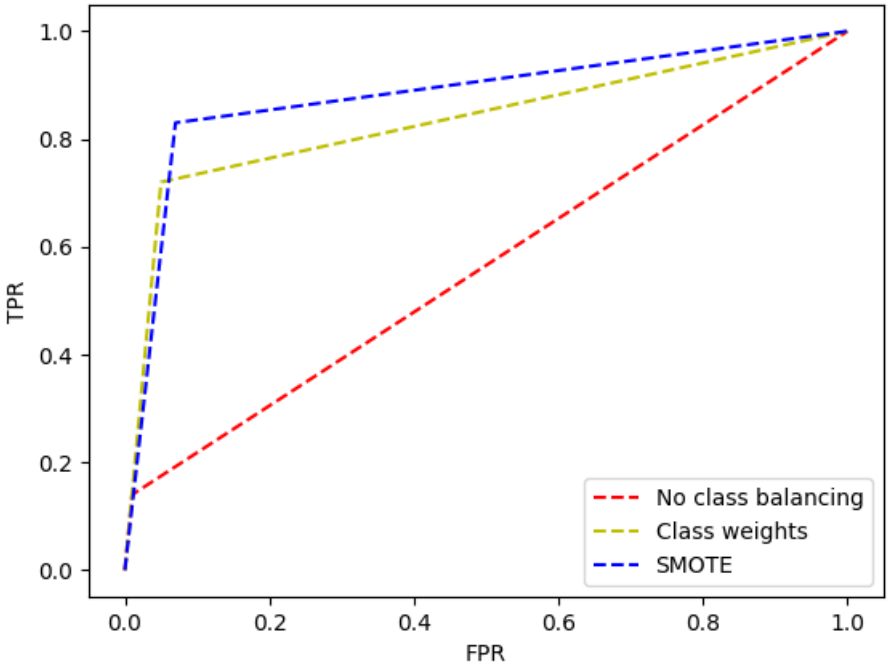
As we can see in table 4.2a, and table 4.2b both corresponding models predict relatively low amounts of top-10 samples. This causes the precision for the trees generated without class balancing techniques to be relatively high.

ROC-curves and AUC

From the quality metrics in section 4.1.1 we can draw up the following ROC-curves.



(a) ROC-curves for decision trees with Gini gain as splitting criterion.



(b) ROC-curves for decision trees with information gain as splitting criterion.

From the ROC-curves in figure 4.1a, and 4.1b we can calculate the following AUC's.

		AUC
Gini gain	No class balancing techniques	0.645
	Class weights	0.835
	SMOTE	0.905
Information gain	No class balancing techniques	0.565
	Class weights	0.835
	SMOTE	0.88

Table 4.2: AUC's for the decision all built decision trees.

As visible in table 4.2 the decision tree that had a SMOTE balanced data set, and Gini gain as splitting criterion has the largest AUC, and is thus our best performing model.

4.1.2 Visualising the decision tree

We now have one decision tree that can predict the successfulness of a cyclist. A visualisation of the tree is visible in figure 4.1.

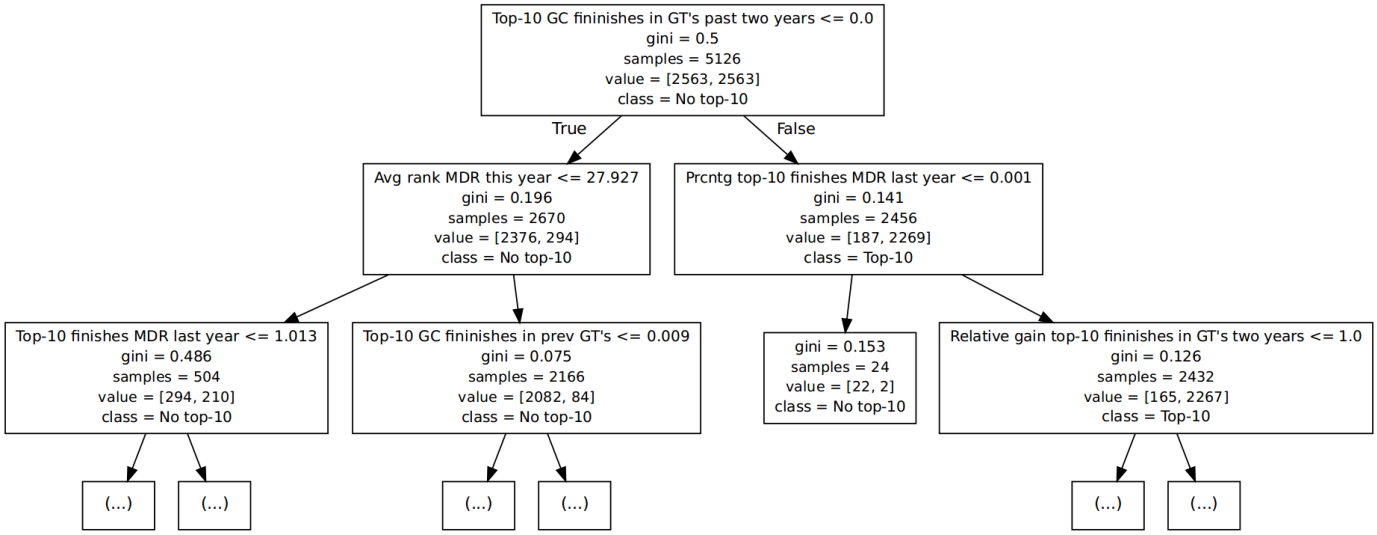


Figure 4.1: Decision tree generated with a SMOTE balanced data set, and Gini gain as splitting criterion. Only the first three levels of the tree are shown because these levels show the most important criterion upon deciding the successfulness of a cyclist. Every node contains five fields, and each field contains metadata about the node. The first field is the attribute test that the algorithm splits on. The second field is the value of the splitting criterion, in that node. The third field is the amount of samples that are present in the node. The fourth field contains a set that represents the distribution of the samples over the target classes. The first element of this set represents the amount of samples that will be classified as no top-10, and the second element of this set represents the samples that will be classified as top-10. The last field contains the class prediction that the node will make, so top-10 or no top-10.

As we can see in figure 4.1 the most important criterion upon deciding the successfulness

of a cyclist is the number of top-10 GC finishes in a GT that a cyclist has obtained during the past two years, because this is the first split in the tree. The tree makes a clear distinction between cyclists that have obtained no top-10 GC finishes in a GT in the past two years, and cyclists that have obtained one or more top-10 GC finishes in a GT in the past two years. When a cyclist does not have a top-10 GC finish in a GT in the past two years, the tree will almost always classify him as no top-10. But when a cyclist has at least one top-10 GC finish in a GT in the past two years, this cyclist will have a very large chance of being classified as top-10.

When we look at the splits after the root node, we see that the percentage of top-10 finishes in MDR's last year is the next criterion for deciding the successfulness of a cyclist. Cyclists that have at least one top-10 finish in an MDR last year will be classified as top-10 while cyclists that do not have a top-10 finish in an MDR last year will be classified as no-top10. This feature has a common property with the feature of the root node, namely that these features are both related to how well a cyclist has performed in the past. If we look one level deeper in the tree, we again see that the tree splits on features that are related to how well a cyclist has performed in the past. We can thus conclude that cyclists that have performed well in the past have a larger chance of being classified as top-10, than cyclists that have performed less in the past.

4.2 Decision trees with modified data set

In section 4.1 we have seen that the first splits in the decision tree were all on features that are related to a cyclist's performance in the past, implying that strong performance in the past predicts the future success of a cyclist. As our data set also contains features about a cyclist's physique, and experience, we would like to know if and how these features contribute to a cyclist's successfulness. In order to investigate the influence of these characteristics we will again build a decision tree, but this time the input data set does not contain features that are related to a cyclist's performance in the past. The resulting input data set contains the following features:

- `Rnk`
- `Age`
- `Weight`
- `Height`
- `BMI`
- `BSA`
- `Total kms last year`
- `Kms this year 3mnths before GT start`
- `RD this year 3mnths before GT start`
- `Years as pro`

- `All time GT starts`
- `GT finishes last year`
- `MDR finishes last year`
- `SDR finishes last year`
- `Days since last MDR`
- `Days sine last SDR`

With this input data set we will perform the same experiments as in section 4.1, in order to find out what the influence of different splitting criterion, and class imbalance is on the performance of our model. Next, we will choose the best performing model, and from this model we can then derive which characteristics contribute to the successfulness of a cyclist.

4.2.1 Quality metrics

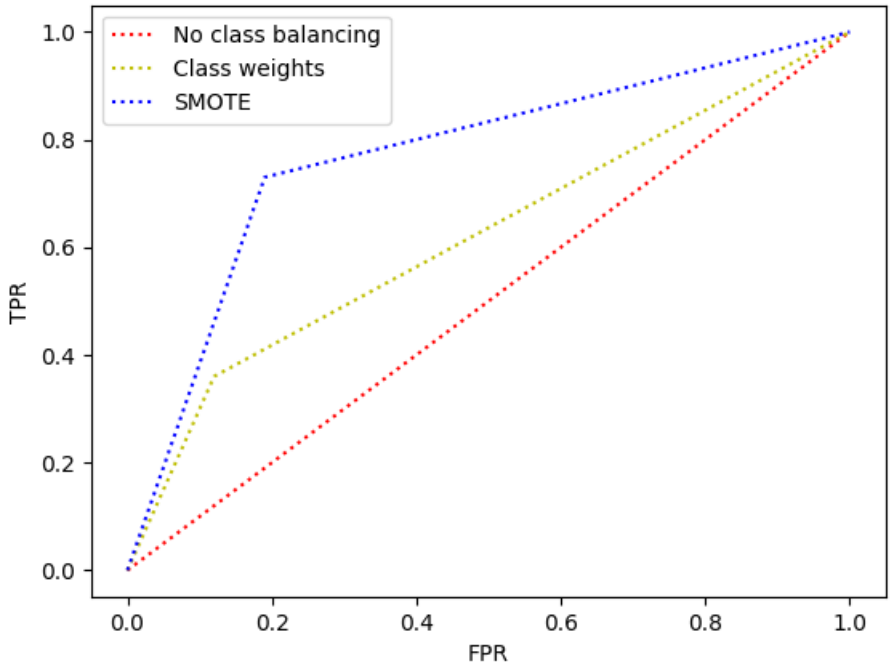
		Precision	Recall	f1-score	FPR
Gini gain	No class balancing techniques	0.00	0.00	0.00	0.00
	Class weights	0.17	0.36	0.23	0.12
	SMOTE	0.20	0.73	0.32	0.19
Information gain	No class balancing techniques	0.50	0.05	0.08	0.003
	Class weights	0.18	0.55	0.27	0.16
	SMOTE	0.16	0.59	0.25	0.2

Table 4.3: Quality metrics from the decision trees built with an input data set that does not contain features related to a cyclist's performance in the past.

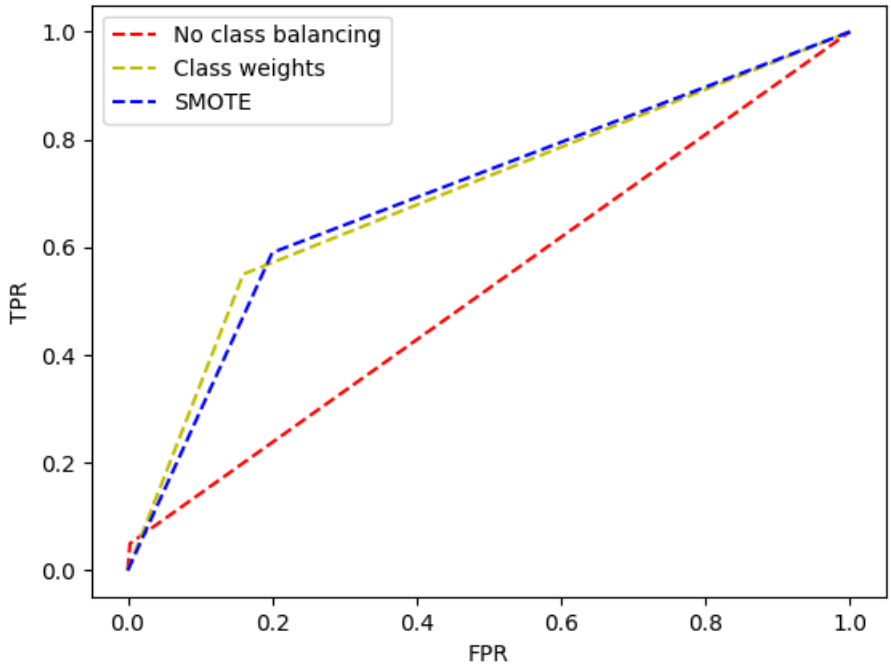
When we look at table 4.3, we can see that the decision trees that have been built with class balancing techniques perform better than trees that have been built without class balancing techniques, implying that class balancing techniques enhance the performance of our model. Furthermore, we can see that the tree that has been generated without class balancing techniques and Gini gain has all quality metrics set to 0.00, because the tree classifies every sample as no top-10. Overall there are no large differences between the trees that have been built with Gini gain, and the trees that have been built with information gain. When comparing the quality metrics in table 4.3 with the quality metrics in table 4.1, we see that the performance for decision trees built with an input data set that does not contain features about a cyclist's performance in the past has dropped.

ROC-curves and AUC

From the quality metrics in table 4.3 we can draw up the following ROC-curves.



(a) ROC-curves from the trees built with Gini gain, and an input data set that does not contain features that are related to a cyclist's performance in the past.



(b) ROC-curves from the trees built with information gain, and an input data set that does not contain features that are related to a cyclist's performance in the past.

From the ROC-curves in figure 4.2a, and figure 4.2b we can calculate the following AUC's.

		AUC
Gini	No class balancing techniques	0.50
	Class weights	0.62
	SMOTE	0.77
Information gain	No class balancing techniques	0.52
	Class weights	0.69
	SMOTE	0.69

Table 4.4: AUC's calculated from the ROC-curves in figure 4.2a, and figure 4.2b

From the AUC scores in table 4.4 we can choose that the tree that has been built with a SMOTE balanced data set, and Gini gain as splitting criterion is again the best performing model. This model has a smaller AUC, and thus less accurate predictions, than the best performing model in section 4.1.

4.2.2 Visualising the decision tree

In order to find out what decisions are made upon deciding the successfulness of a cyclist, we have to visualise the decision tree.

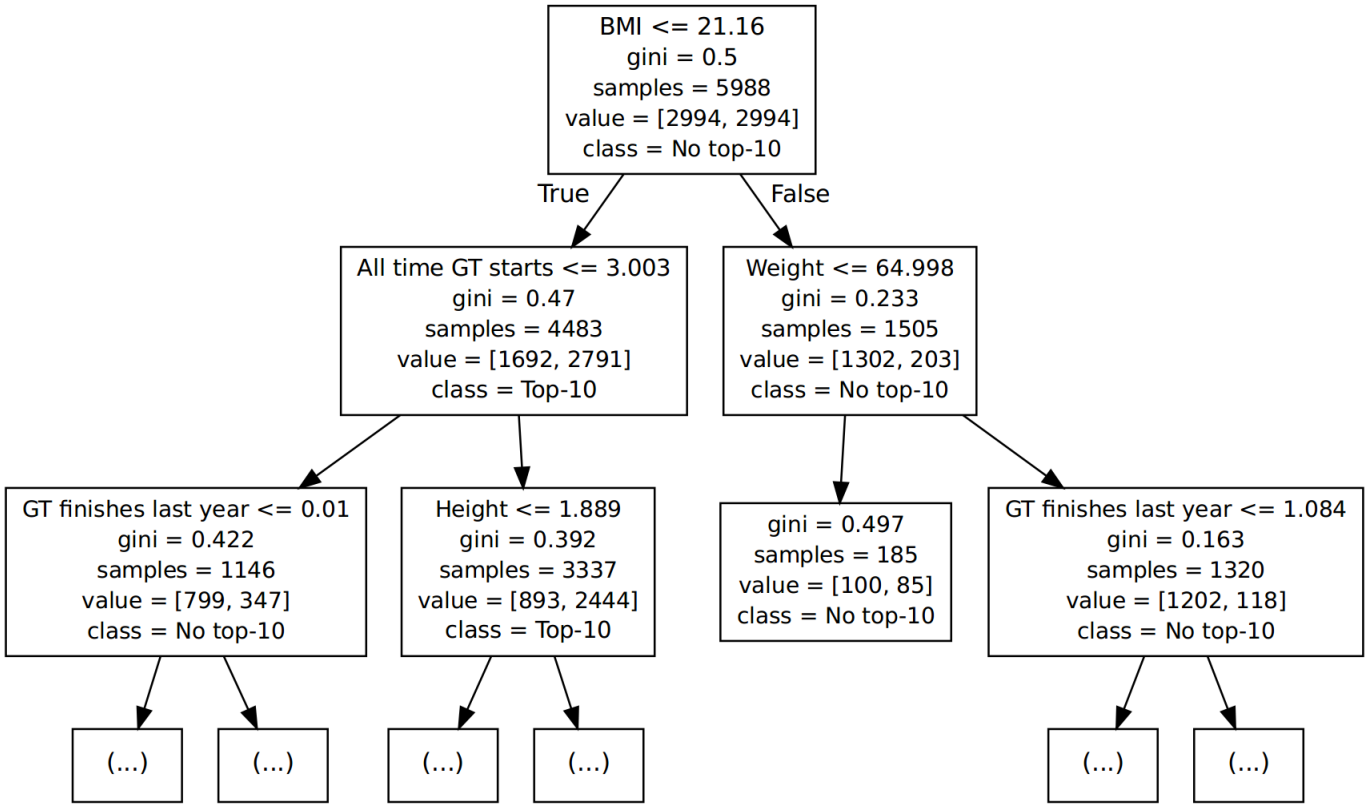


Figure 4.2: Decision tree built with a data set that does not contain the features that are related to a cyclist's performance in the past. Only the first three levels of the tree are shown as these levels show the most important criteria upon deciding the successfulness of a cyclist.

As we can see in figure 4.2 the most important criterion upon deciding the successfulness of a cyclist when not accounting for previously achieved results, is the value of a cyclists BMI, as this is the first split in the tree. Cyclists that have a relatively low BMI value (≤ 21.16) have a large chance of being classified as top-10 while cyclists that have a relatively high BMI value will always be classified as no top-10. If we look closer at the tree we see that the cyclists that have a relatively low BMI, and relativeley high level of experience in GT's (≥ 3 all time starts in a GT) have an even larger chance of being classified as top-10. We can also see that the cyclists that have a relatively low BMI, but do not have a relatively high level of experience in GT's will be classified as no-top10. The tree thus shows us that in order to be successful, a cyclist needs a relatively low BMI value, and a relatively high level of experience in GT's.

Chapter 5: Discussion

In chapter 1 we stated our research question as: *What are the characteristics of successful cyclists in Grand Tours?* From the decision tree built in section 4.1 we have learned that the most important characteristic upon deciding the successfulness of a cyclist is his performance in previous Grand Tours. We have seen that achieving just one top-10 GC finish in a GT distinguishes the cyclists that will be classified as top-10 from the cyclists that will be classified as no top-10. If we do not take previously achieved results into account we see that the performance of our model drops. The resulting decision tree in figure 4.2 shows us that cyclists that have a relatively low (≤ 21.16) BMI value, and a relatively large level of experience (≥ 3 all time GT starts) will be classified as top-10.

The fact that cyclists that have performed well in the past are likely to perform well in the future can be seen as intuitive, but has also been proved for athletes in different sports disciplines such as tennis [ID14]. Furthermore, previous research has shown that modern day GT winners weigh around 70 kilograms and are approximately $1,80$ meters tall [AL01]. This comes down to a BMI value of 21.6 which is close to the BMI value that we have found for successful cyclists.

The implications for coaches and cyclists of these findings are two-sided. On one hand the results are discouraging, and show that a cyclist needs to have been successful in order to become successful. This is a characteristic that can not be trained, or enhanced and is in accordance with Niednagel's findings that an athlete's success is based on predetermined factors [Nie92]. On the other hand, the findings show that cyclists with a relatively low BMI value, and large level of experience in GT's also have a large chance of being successful. Although the these findings are less accurate, the results show that success may be influenced by adaptable factors.

The accuracy of our findings could have been enhanced if we had acquired more data about a cyclist's physical profile. Existing research has shown that there exists substantial differences in a cyclist's maximum oxygen intake (VO_{2max}), or maximum power output (W_{max}) between different rider types. Taking these characteristics into our model might have enhanced the performance of our model.

In this thesis we have also looked into how different splitting criteria have influenced the performance of our model. Our results show that there exists very little difference in performance between decision trees with Gini gain, or information gain as splitting criteria. This is in agreement with other studies that have shown that there is no splitting criteria that is consistently superior to others [BB78], and that Gini gain and information gain make different splits in only 2% of the presented cases [RS04b].

Furthermore, we have also examined the influence of class imbalance on the performance of our model. We have found that class imbalance causes the model to learn only the majority class of the data set, causing poor performance. This has been shown in plenty of other studies [JS02]. In order to overcome this problem, we have used SMOTE and class weights, as current research has shown that both methods are effective at countering class imbalance. Our results in section 4.1.1, and section 4.2.1 have shown that applying SMOTE, or class weights results in a decision tree that has better performance than a decision tree that has been built without class balancing techniques. The trees that have SMOTE applied slightly outperform the trees that have class weights applied. Currently there exists no research that states which of these class balancing techniques causes consistently superior performance.

Chapter 6: Conclusion

The research question of this thesis is: "What are the characteristics of successful cyclists in Grand Tours?" In order to give an answer to this question we have collected a data set containing features about a riders' experience and previously achieved results as well as some features about a riders physique. We have used Gini gain, and informatino gain as splitting criterion for our decision tree, and found that there exists very little difference in performance when using either one of these criterion.

Furthermore, we have seen that class imbalance deteriorates the performance of our model. In order to overcome this, we have applied SMOTE, and class-weights, with SMOTE performing slightly better. From the decision tree with the most accurate predictions, we can derive that the number of top-10 GC finishes in GT's in the past two years is the most important characteristic upon deciding the successfulness of a cyclist. If we do not include previously achieved results in our data set, we see that the performance of our model drops and that BMI value, and level of experience in GT's determine the successfulness of a cyclist.

6.1 Future work

This research can be expanded by adding more features to the data set. Interesting features to include would be more physical characteristics of a cyclist. Such as maximal oxygen intake, maximal heartbeat rate, or average power output. These characteristics tell a lot about a cyclist's performance, and could lead to a very accurate prediction model. Other features that can be added are features related to a riders training scheme such as number of altitude training days, or training kilometers. These features give information about a riders preparation for a GT.

Another interesting niche to look at are the psychological characteristics of a rider. Examples of this are did the rider see his family during a GT? How much sleep did a rider get during a GT? Did the rider speak to the press during the GT? Adding all these features would give a much broader profile of a rider, and allow for a much more in depth analysis of the characteristics of successful cyclists in Grand Tours.

Bibliography

[AL01] Jose L. Chicaro Alejandro Lucia, Jesus Hoyos. Physiology of professional road cycling. *Sports Med*, 2001.

[BB78] M. Ben-Bassan. Myopic policies in sequential classification. *IEEE Transactions on Computing*, 1978.

[dBdB16] Delafied du Bois and Eugene du Bois. A formula to estimate the approximate surface area if height and weight be known. *Clinical Calorimetry*, 1916.

[DR13] M. Durairaj and V. Ranjani. Data mining applications in healthcare sector: A study. *International Journal of Scientific Technology Research*, 2013.

[Eri07] K. Anders Ericsson. The influence of experience and deliberate practice on the development of superior expert performance. *The cambridge handbook of expertise and expert performance*, 2007.

[GG05] Auroop R Ganguly and Amar Gupta. Data mining technologies and decision support systems for business and scientific applications. *ENCYCLOPEDIA OF DATA WAREHOUSING AND MINING*, 2005.

[Hoa91] Deborah G. Hoare. Physiological characteristics of elite judo athletes. *International journal of Sports medicine*, 1991.

[ID14] Seppo Isoahola and Charles Dotson. Psychological momentum: Why success breeds success. *Review of General Psychology*, 2014.

[Jeu01] Asker E. Jeukendrup. Improving cycling performance. *Sports medicine*, 2001.

[JS02] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 2002.

[Koh95] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, 1995.

[Luo08] Qi Luo. Advancing knowledge discovery and data mining. *WKDD*, 2008.

[McC10] John McCullagh. Data mining in sport: A neural network approach. *International Journal of Sports Science and Engineering*, 2010.

[Nar19] Sarang Narkhede. Roc auc, 2019. [Online accessed July 31, 2019].

[NC02] L.O. Hall W.P. Kegelmeyer N.V. Chawla, K.W. Bowyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002.

[Nie92] J. Niednagel. *Your key to sports success*. 1992.

[nMP01] Iñigo Mujika and Sabino Padilla. Physiological and performance characteristics of male professional road cyclists. *Sports Medicin*, 2001.

[Ofo13] Bahadorreza Ofoghi. Data mining in elite sports: A review and a framework. *Measurement in physical education en exercise science*, 2013.

[Ras13] Hamid Rastegari. A review for data mining techniques for result prediction in sports. *Advances in Computer Science*, 2013.

[RB19] Fadi Thabath Rory Bunker. A machine learning framework for sport result prediction. *Applied computing and informatics*, 2019.

[Ren61] Alfred Renyi. On measures of entropy and information. In *Fourth Berkeley Simposium*, 1961.

[Ros16] Karl Rosaen. 10-fold cross validation, 2016. [Online; accessed July 26, 2019].

[Roz17] R. Rozendaal. Modeling performance of elite cyclists. Master's thesis, TU Delft, 2017.

[RS04a] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 2004.

[RS04b] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 2004.

[SP99] G. Cuesta J. Goiriena S. Padilla, I. Mujika. Level ground and uphill cycling ability in professional road cycling. *Medicine Science in Sports*, 1999.

[Spi07] Martin Spieler. Predicting athletic success: Factors contributing to the success of ncaa division i aa collegiate football players. *Athletic insight*, 2007.

[TN89] R. Schall T.D. Noakes, K.H. Myburg. Peak treadmill running velocity during the vo2 max test predicts running performance. *Journal of Sports sciences*, 1989.