Master thesis Business Analytics

Ranking the Future Stars

A machine learning approach to predict the performance
of a potential Formula 3 driver.

Claudia Sulsters

August 30, 2018

Vrije Universiteit Amsterdam
Faculty of Science
De Boelelaan 1081a
1081 HV Amsterdam

Supervisors VU:

Prof. dr. S. Bhulai
Dr. M. Hoogendoorn

Van Amersfoort Racing B.V.
Edisonweg 2
3899 AZ Zeewolde

Supervisors VAR:

Rik Vernooij
Peter van Leeuwen

Preface

This thesis is written to conclude my master Business Analytics at the Vrije Universiteit Amsterdam. Business Analytics is a multidisciplinary program, aimed at improving business processes by applying a combination of methods based on mathematics, computer science, and business management. The master's degree is concluded with a six-month internship at a company. During my internship at Van Amersfoort Racing, I had the opportunity to combine my passions for auto racing and data science in a challenging graduation project.

One might think that it is surprising to conclude the master Business Analytics at a racing team. However, given my passion for Formula auto racing and my interest in applying statistics to sports data, no other internship would be more fitting for me. My enthusiasm for Formula auto racing rapidly grew after my first actual visit to a Formula 1 race, the 2017 Spanish Grand Prix. I was impressed by the fast Formula 1 cars driving around the corners of the circuit de Catalunya. Formula 2 and Formula 3 races are often even more interesting to watch since young, talented drivers are competing with each other to achieve their greatest dream: becoming a Formula 1 driver. Van Amersfoort Racing helps young drivers build successful careers in auto racing but only for a few this dream will become true one day. My graduation project was focused on finding 'these few', most talented, drivers. Also, I had the opportunity to get a glimpse of how a Formula auto racing team operates and how dedicated and passionate the team educates the young drivers.

I would like to thank the Van Amersfoort Racing team for giving me the opportunity to work on this challenging graduation project. In particular, I would like to thank my VAR supervisors Rik Vernooij and Peter van Leeuwen and my VU supervisor Sandjai Bhulai for the advice and guidance throughout the graduation project.

Summary

The goal of this research is to support Van Amersfoort Racing in the driver selection process by providing an objective method to select talented drivers. This method uses machine learning techniques to predict the driver performance in the FIA F3 European Championship based on a driver's career path, i.e., his/her performance in other relevant championships of single-seater auto racing and karting. Then, rankings of potential F3 drivers are produced based on predicted driver performance rather than actual championship results. Van Amersfoort Racing can use these rankings to decide which drivers are most talented, and, thus, should be contracted for the next season. Based on this problem statement, we formulated the following research question:

How accurately can we predict the performance in the FIA F3 European Championship based on a driver's career path?

We have used a machine approach to answer the above research question. First, we have collected data containing race results from the FIA F3 European Championship, other relevant championships of single-seater auto racing and kart championships (2008-2017). These data are then used to create features that describe a driver's career path to the FIA F3 European Championship. The machine learning models try to learn the relationship between the features, the performance in other championships, and the driver performance in the FIA F3 European Championship. Challenges of the machine learning task are the large amount of missing values in the data set and the large number of features. We thus have chosen machine learning models that can handle missing values and that facilitate the feature selection process. These models are the regression tree, the random forest model, and the gradient boosting model. We have split the data set in a training set (80%) and a test set (20%). The training set is used to train the models and optimize the hyperparameters, while the test set is used to evaluate the performance. Moreover, we evaluated the performance of the models on different test sets in a bootstrap procedure to obtain a more reliable measure of the overall performance. The performance was measured using the root-mean-square error (RMSE), Spearman's rank correlation coefficient, and the normalized discounted cumulative gain (nDCG). The results have shown that both the gradient boosting model and the random forest model outperform the regression tree using any reasonable significance level, which was expected at beforehand. Boosting and bagging techniques are namely known to improve the predictive performance by combining a large number of regression trees. Finally, we concluded that the gradient boosting model performed significantly better than the random forest model using a significance level of 0.1.

The gradient boosting model was chosen as the final model and used to analyze the importance of the features. The features *Team* and *Nationality* are considered important by the gradient boosting model for predicting the performance in the FIA F3 European Championship. Moreover, the performance during previous seasons of the FIA F3 European Championship is important. Other championships that are important are GP3 Series, British Formula 3 International Series, Eurocup Formula Renault 2.0, British Formula Tree championship, Formula Renault 2.0 NEC, and CIK FIA European Championship KF (karting). We also used the final model to rank the drivers of the FIA F3 European Championship 2018. This ranking was compared to the standings of the FIA F3 European Championship to analyze the quality of the predictions.

We can conclude that this research provides a new approach for finding talented drivers in single-seater auto racing. The machine learning model can support Van Amersfoort Racing in the driver selection process by providing them with rankings based on driver performance. These rankings can help Van Amersfoort Racing to find a new driver for the next season. We used quantitative as well as qualitative measures to evaluate the performance of the machine learning models and to answer the research question. The gradient boosting model performs best and seems to produce usable predictions. However, its performance should be carefully evaluated in the future when it is used to find new drivers. The model should be used as a supporting rather than a guiding tool and future predictions should be compared with expert opinions.

Contents

1 Introduction

1.1 About Van Amersfoort Racing

Van Amersfoort Racing is a Dutch racing team that was founded by Frits van Amersfoort in 1975. The team competes in the FIA Formula 3 (F3) European Championship and the ADAC Formula 4 (F4) Championship. Van Amersfoort Racing helps young drivers build successful careers in auto racing. The drivers that join van Amersfoort are supported by a team of professional race engineers and driver coaches, including Frits van Amersfoort himself. The company's headquarters are located in Zeewolde and have three in-house simulators with existing race tracks available. Van Amersfoort Racing educates the young drivers by letting them practice their driving skills during simulator training as well as on the track during the race weekends. Under supervision of race engineers, the young drivers not only learn how to drive fast laps, but, more importantly, to understand the dynamics of the car. Their data is benchmarked to data of the best racing drivers in order to understand how their driving can be improved. Over the years, many successful drivers have been part of the van Amersfoort Racing team such as current Formula 1 (F1) drivers Max Verstappen and Charles Leclerc.

1.2 Research question

A challenge that is yearly faced by Van Amersfoort Racing is concerned with which drivers they should contract for the next season. Contracting talented drivers is the key to success in F3 and F4 since driver performance is often more important than car performance. However, it is not always straightforward to decide which driver is most talented. Drivers cannot easily be compared because they compete in different championships of single-seater auto racing or within the same championship but in different teams. Championship results, such as finishing position and championship points scored, are thus influenced by team effects as well as competition effects. Comparisons that are based on championship results but do not take the team and competition effects into account are thus unfair comparisons in terms of driver performance. However, most racing teams contract their drivers based on data about championship results and experts opinions, rather than on objective measures of driver performance.

The goal of this master thesis is to support Van Amersfoort Racing in the driver selection process by providing an objective method to select talented drivers. This method uses machine learning techniques to predict the driver performance in the FIA F3 European Championship based on a driver's career path, i.e., his/her performance in other relevant championships of single-seater auto racing and karting. Then, rankings of potential F3 drivers are produced based on predicted driver performance rather than actual championship results. Van Amersfoort Racing can use these rankings to decide which drivers are most talented, and, thus, should be contracted for the next season.

Based on this problem statement, we can formulate the following research question:

How accurately can we predict the performance in the FIA F3 European Championship based on a driver's career path?

1.3 Research approach

We use a machine learning approach to answer the above research question. The structure of this thesis follows the common structure of any other machine learning project, as shown in Figure 1.

The first step is to define and understand the problem from a business perspective. We already defined the problem statement above and we formulated the research question accordingly. Another important part of problem understanding is reviewing related problems and relevant literature. Section 2 provides an overview of relevant literature in the field of ranking athletes based on certain performance measures. In particular, methods that are used in the literature to rank F1 drivers are discussed. The second step is

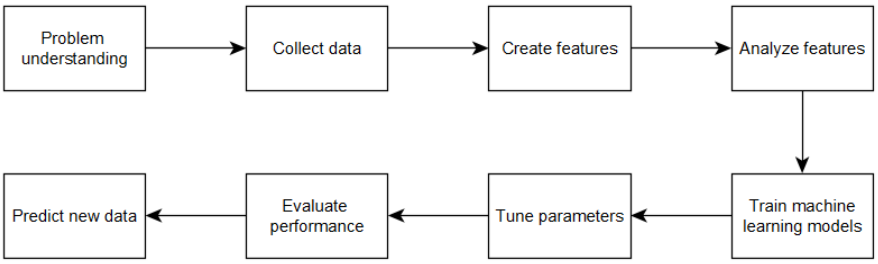


Figure 1: Research approach; common structure of a machine learning project.

to collect data that can be used to train the machine learning models. Since there is no ready-to-use data set available, we have to collect our own data. Section 3 discusses how the data is collected, transformed, and cleaned. These data can then be used to create features that describe a driver's career path up to the FIA F3 European Championship. These features contain the performance of the drivers in other relevant championships of single-seater auto racing and karting. The feature engineering process is described in Section 4. This section also presents the statistical model that is used to distinguish between driver performance, team performance and competition effects. We use the results of the statistical model to create features that describe the driver performance in certain championships and to create the response variable, the driver performance in the FIA F3 European Championship. All features are analyzed on their expected predictive power in Section 5. Now that we have a data set available for the machine learning models to learn from, we have to choose models that are able to perform the machine learning task at hand. These machine learning models will try to learn the relationship between the features, the performance in other championships, and the driver performance in the FIA F3 European Championship. The machine learnings task at hand is described in Section 6 as well as the theoretical background and the implementation of the machine learning models. Each machine learning model relies on a certain set of hyperparameters that needs to be tuned to improve the performance. The method that is used for parameter tuning is also described in Section 6. The performance of the machine learning models is evaluated in Section 7. The model that performs best is chosen as our final model. We use the final model to analyze the importance of the features. Moreover, we provide predictions for unseen data by pretending that we have to find 2018 drivers in the winter of 2017. The resulting ranking is compared with the standings of the FIA F3 European Championship at this moment. Finally, the results are discussed and conclusions are drawn in Section 8 and Section 9, respectively.

2 Literature review

This section discusses some relevant literature that is available in the field of ranking athletes. We discuss three methods that are proposed to rank F1 drivers and compare these. Finally, we will argue why these methods do not provide an adequate answer to the research question and, thus, how this research will attribute knowledge to the field of ranking drivers.

Many statistical models have been proposed in the literature to rank athletes in a variety of sports. Berry, Reese, and Larkey (1999) addressed the problem of comparing abilities of players from different eras in three professional sports: hockey, golf, and baseball. They compared players whose careers took place in different eras using that some players in the data set had overlapping careers. Aitken (2004) analyzed the relevance of outliers in such comparisons and stated that outliers indicate extraordinary athlete performances. Other literature has focused on a particular area of sports such as basketball (P. Kvam & Sokol, 2006; West, 2006; Brown & Sokol, 2010), football (West & Lamsal, 2008; Mease, 2003), cricket (Davis, 2000), baseball (P. H. Kvam, 2011), and skiing (Glickman & Hennessy, 2015). All these sports are organized in such a way that multiple teams or players compete against each other simultaneously in one event or a series of events. The outcome of such an event is often a ranked order of the competitors. Therefore, these studies often used similar methods to obtain athlete rankings including maximum likelihood ranking, (linear) regression techniques or Markov chains.

However, applying such methods to F1 or to auto racing sports, in general, is more complex. According to Phillips (2014), this is caused by the unusual structure of the data, which include scoring finishes, a variety of types of non-scoring finishes and non-finishes. Another complexity that is mentioned by Phillips is that no straightforward performance metric exists to measure driver performance because the performance is strongly influenced by team, car, and competition effects. As a result, finishing positions or championship points scored are considered to be unreliable measures of driver performance.
Nevertheless, methods do exist that rank F1 drivers based on some performance metric. Anderson (2014) used maximum likelihood ranking to rank drivers from the 2012 F1 season. Eichenberger, Stadelmann, et al. (2009), Phillips (2014) and Bell, Smith, Sabel, and Jones (2016) tried to reveal the best F1 driver of all-times by fitting (non-)linear regression models. These methods differ on the structure of the model (fixed effects vs. random effects), on the performance metric that is used, on how non-finishes are handled and on the variables that are included in the model. We will now discuss these methods in more detail.

According to Anderson (2014), rankings based on the number of championship points scored in a series of races, also called point-based rankings, often fail to provide an accurate ranking of driver performance. One reason is that drivers often compete in a different number of races so that the total number of championship points scored by two different drivers cannot be compared. Second, the level of competition varies per race and, third, the ranking is dependent on the number of championship points assigned to each finishing position. For the latter, it is unclear which point scale would produce the ranking that most accurately reflects the relative driver performance. Thus, one might say that point-based rankings are dependent on a subjective points scale. Rankings based on the average number of championship points per race can be used as an alternative to solve the problem that drivers compete in a different number of races but the other two problems remain.
Anderson presents three alternative models that can be used to rank F1 drivers and overcome the shortcomings of point-based rankings. Two methods are based on paired-comparisons among the drivers and can be estimated using common binary-choice regression methods. These paired-comparison models compare each pair of drivers and quantify which driver outperforms the other based on the race results. The other method is based on the rank-ordered logit model and computes the likelihood of a certain finishing order. Drivers that do not finish are handled as if they did not participate in the race. According to Anderson, it is unlikely that the models can correctly account for mechanical problems or crashes because of the underlying distributions of driver performance they assume. The models could be adjusted to account for non-finishes such that the driver performance reflects that fact that a driver is more likely

to crash or suffer from a mechanical failure. However, it is difficult to draw reliable conclusions about the non-finishing probability based on the small number of events.
A disadvantage of these maximum likelihood methods is that they cannot quantify the influence of team performance and the level of the competition on the performance measure. The assumption is made that valid comparisons can be made between any two drivers based on the results of a race which is for auto racing sports not true in general. Only considering paired-comparisons or the finishing order is insufficient to adequately determine the driver performance because the influence of team and competition effects is neglected. The models of Eichenberg and Stadelmann, Phillips and Bell et al. do not have this disadvantage because they provide estimates for team and competition effects.

Eichenberger et al. (2009) were the first to propose an objective method to rank all-time F1 drivers by separating driver performance from car performance. They used a linear model to compute driver performance estimates for each F1 driver based on a data set from the start of F1 racing in 1950 up to 2006. The finishing position was chosen as dependent variable in the model and driver performance was assumed to be constant over a driver's career. The model includes driver effects and car-year effects and contains control variables for the number of drivers finishing the race, technical dropouts, weather conditions and the race distance. The model was thus specified as follows:

$$y_{it} = \alpha_i + \gamma_{s,i} + X\beta + u_{it} \tag{1}$$

where

$$\begin{aligned} \alpha_i &= \text{dummy variable capturing performance of driver } i, \\ \gamma_{s,i} &= \text{dummy variable representing car-year-specific effects,} \\ X &= \text{matrix containing the other control variables,} \\ \beta &= \text{coefficient corresponding to the matrix } X, \text{ and} \\ u_{it} &= \text{error term.} \end{aligned}$$

Non-finishes are divided into 'human dropouts' and 'technical dropouts'. 'Human dropouts' can, for example, be caused by accidents, collisions or disqualification, while 'technical dropouts' are due to technical failures of the car. Eichenberg and Stadelmann control for technical dropouts in their model and compute for human dropouts a hypothetical classification that is always worse than achieving a classification. Finally, they establish a world championship ranking based on the average driver performance across a driver's career.

Phillips (2014) used championship points instead of finishing positions as dependent variable in his model. He argued that points are a more reliable metric than positions because of the following reasons. First, the model becomes sensitive to bad results when the finishing position is used as dependent variable. For example, a driver that finishes first several times and then finishes last (e.g., due to a technical failure) may have a worse performance than a driver that consistently finishes third. Second, drivers that regularly finish in high positions are penalized more by a non-finish than drivers that often finish on lower positions. Third, drivers that often finish in high or low positions have less variability in finishing positions than drivers that finish in the middle of the field. Phillips addresses this problem by accounting for nonlinear changes in the performance metric (championship points) as a function of driver and team performance. A disadvantage of using championship points instead of finishing position is that traditional points systems cannot discriminate between non-scoring positions. This problem was addressed by Phillips by using an extended points system that also assigns (fractional) points to non-scoring positions. Moreover, he classified non-finishes as non-driver failures or driver-failures and removed all non-driver failures from the analysis.
The model that is used by Phillips assumes that the performance variable y_{ijk} is a linear function of driver, team, and competition effects on performance

$$y_{ijk} = \alpha_i + \beta_j - \delta_k + \epsilon_{ijk}, \tag{2}$$

where

$$\begin{aligned}
\alpha_i &= \text{fixed effect representing driver performance,} \\
\beta_j &= \text{fixed effect representing team performance,} \\
\delta_k &= \text{fixed effect representing difficulty scoring in season } k, \\
&\quad \text{due to competition with other drivers, and} \\
\epsilon_{ijk} &= \text{random effect representing variability in performance.}
\end{aligned}$$

Using this model, driver and team contributions to performance as well as the effects of competition with others were estimated and the underlying driver performances were revealed. The performance variable y_{ijk} can be used to compute the average number of championship points scored by a driver. When the performance variable is adjusted for the team and competition effects, one can compute an adjusted average number of championship points solely based on driver performance. Finally, F1 drivers from 1950-2013 were ranked based on 1-year, 3-year, and 5-year peak performances.

The most recent attempt to rank F1 drivers was done by Bell et al. (2016). Bell et al. used multilevel models or random-coefficient models to find rankings of all-time F1 drivers conditional on team performance. Moreover, they tried to answer the question whether teams or drivers are more important in F1. This is done by quantifying the influence of team and driver effects on the overall performance and evaluating how these effects vary over time and under different racing conditions. The number of championships points scored by drivers in a race is used as the response variable in a non-linear model that partitions variance into team, team-year, and driver levels. The most important difference between the method of Bell et al. (2016) and the methods of Eichenberger et al. (2009) and Phillips (2014) is that they used multilevel models or random effects models rather than fixed effects models to represent drivers and teams. Bell et al. argued that this multilevel structure of their model allows them to treat non-finishes as non-scoring finishes, regardless of the cause. This is because the multilevel structure will automatically assign team/car failures to team or team-year levels and driver failures to driver-levels. Moreover, the multilevel model allows Bell et. al to include team levels as well as team-year levels to model some relationship between team performance across different years while the other models assume independent team performances for different years. Finally, another advantage of the multilevel model is that data from all drivers and teams in history can be used while the other models were restricted to a subset of the data. Eichenberger and Stadelmann only included F1 drivers with at least one championship point, while Phillips only included drivers that finished at least 3 races in a year.

The multilevel model that is used by Bell et al. is the following

$$\text{Rankit(Points)}_i = \beta_0 + \beta_1 \text{Ndrivers}_i + \beta_2 \text{Comp}_i u_{\text{Driver}} + v_{\text{Team}} + w_{\text{TeamYear}} + e_i, \tag{3}$$

$$\begin{aligned}
u_{\text{Driver}} &\sim N(0, \sigma_u^2), \\
v_{\text{Team}} &\sim N(0, \sigma_v^2), \\
w_{\text{TeamYear}} &\sim N(0, \sigma_w^2),
\end{aligned}$$

where σ_u^2, σ_v^2, and σ_w^2 summarize the between-driver, between-team, and between-team-year variance respectively, and σ_e^2 summarizes the within-race variance net of driver, team and team-year characteristics. As can be seen in (3), the rankits of the championship points scored is used as the response variable. Rankits of a data set are the expected values of the order statistics of a sample from the standard normal distribution with the same size as the data. The variable *Ndrivers* is used to resolve the problem that drivers that compete against fewer other drivers are assigned a higher performance because the average points scored will be higher. This variable thus controls for the number of drivers in a given race. The variable *Comp* is used to control for the competitiveness of the race. Also, the model is further extended

to include other variables as year, weather and track type in the variance functions. Finally, the model of Bell et al. provides a ranking of the best all-time F1 drivers.

The methods proposed by Eichenberger et al. (2009), Phillips (2014), and Bell et al. (2016) can be applied to race results data from the FIA F3 European Championship and other championships of single-seater auto racing to estimate driver performance by correcting for team and competition effects. This is due to the structure of race results data from lower championships of single-seater auto racing being similar to the structure of race results data from F1. However, any results obtained by these methods insufficiently answer the research question stated in this research because most of the potential F3 drivers did not yet compete in the FIA F3 European Championship. As a result, race results data from the FIA F3 European Championship are missing for most of the potential F3 drivers and cannot be used by the model to estimate driver performance. This research uses machine learning methods to predict the performance of a potential driver in the FIA F3 European Championship based on the performance in other relevant championships of single-seater auto racing and karting. Potential F3 drivers are then ranked based on their predicted performance in the FIA F3 European Championship rather than based on their actual performance.

3 Data

This section presents the data that is used in this research. First, we will discuss the data collection process. The data should contain race results from the FIA F3 European Championship, other relevant championships of single-seater auto racing and kart championships (2008-2017), such that we can reconstruct the career path of (previous) F3 drivers. Second, we will discuss how the data are transformed and cleaned before creating features for the machine learning models.

3.1 Data collection

The data are collected from open-source webpages such as Wikipedia and driverdb.com and consist of race results from the FIA F3 European Championship, other relevant championships of single-seater auto racing and kart championships (2008-2017). The championships that are used to collect data from are chosen in consultation with Van Amersfoort Racing and can be found in the Appendix. Data about race results in single-seater auto racing championships can be found on Wikipedia, while data about kart championships are only available on driverdb.com. Figure 3 shows the race results data of the 2017 FIA F3 European Championship as collected from Wikipedia.

Data are collected from Wikipedia using the R package *rvest* (Wickham, 2016), which facilitates 'harvesting' or scraping the HTML or XML content from webpages. This package also contains functions for extracting the content of an HTML table, which can be used to extract the content of the table shown in Figure 3. The rows of this table represent the drivers that competed in the 2017 FIA F3 European Championship, while the columns contain, for each driver, the position in the championship (*Pos.*), the results (finishing positions) achieved in each round of the championship and the total number of championship points scored (*Points*). The championship consisted of 10 rounds in 2017, where each round again consists of 3 different races. The total number of championship points is equal to the sum of the championship points that a driver scored in each race based on his final position. However, as can be seen in Figure 3, not all drivers participated in all 30 races and some drivers retired from a race. The legend in Figure 2 shows the diversity of non-scoring finishes and non-finishes that is characteristic of auto racing data. The blue results represent non-scoring finishes, while the purple, red, black and white results represent non-finishes. Non-scoring finishes occur because only the first ten finishing positions are awarded with championship points or because it concerns a (guest) driver that is ineligible to score championship points. Non-finishes are for example caused by driver retirements, disqualification, withdrawal or because the driver is not able to appear at the start of the race due to mechanical problems. The methods that we use to handle non-scoring finishes and non-finishes will be discussed in Section 3.2.

Colour	Result
Gold	Winner
Silver	2nd place
Bronze	3rd place
Green	Points finish
Blue	Non-points finish
	Non-classified finish (NC)
Purple	Retired (Ret)
Red	Did not qualify (DNQ)
	Did not pre-qualify (DNPQ)
Black	Disqualified (DSQ)
White	Did not start (DNS)
	Withdrew (WD)
	Race cancelled (C)
Blank	Did not participate (DNP)
	Excluded (EX)

Figure 2: Results that can occur in auto racing data.

Data about the driver-team combinations can also be collected from Wikipedia.com, as can be seen in Figure 4. This figure shows the driver-team combinations of the 2017 FIA F3 European Championship. As can be seen in the *Rounds* column, some drivers did not compete in all races or switched teams during the season. For the FIA F3 European Championship, the chassis and engine of the car are also known for each driver. The status indicates whether the driver was a guest driver, meaning that the driver is ineligible for championships points, or a rookie driver, meaning that this is the driver's first season in the championship. Finally, the *No.* column contains a driver's racing number, but this is not a driver's unique identifier because it can change from season to season.

The driver-team data are used to extend the race results data with information describing in which team each driver participated during which rounds of the championship. For example, David Beckmann was racing for Van Amersfoort Racing during rounds 1-3 and for Motopark during rounds 4-10. Then this information is added by creating one column for each team and one column for the corresponding rounds. Four columns are thus added to the race results data because no driver participated in more than two different teams during the 2017 FIA F3 European Championship.
This method requires that the driver-team data can easily be merged with the race results data based on the driver's name. However, this is not always the case in practice. During the merging process, many inconsistencies in the spellings of driver names were encountered and most of these inconsistencies needed to be solved manually. This difficulty should be taken into account in the future when additional data are collected.

Kart championship data are collected from driverdb.com as these data are not available on Wikipedia.com. The data from driverdb.com are less detailed than the data from Wikipedia and contain championship standings rather than race results. Collecting data from driverdb.com is a complicated process because the content on this website is dynamically created via javascript and cannot be scraped using the *rvest* package. We use another R package, *splashr* (Rudis, 2018), which can be used to scrape dynamic websites, to collect the content from the championship standings tables on driverdb.com. These tables consist of driver names, driver nationalities, the position in the championship, team names, engine, car, and tire specifications and the number of championship points. However, data about the team names and engine, car and tire specifications are often missing.

Pos.	Driver	SIL			MNZ			PAU			HUN			NOR			SPA			ZAN			NÜR			RBR			HOC			Points
1	Lando Norris	1	9	3	1	2	2	2	2	Ret	8	14	3	11	1	3	1	Ret	1	1	3	1	1	2	1	4	2	17	2	11	4	441
2	Joel Eriksson	4	1	2	4	1	4	1	Ret	5	10	2	1	4	10	7	9	2	2	2	12	12	10	9	8	2	1	1	1	4	2	388
3	Maximilian Günther	3	4	4	7	4	3	3	1	1	1	6	6	1	3	2	3	3	Ret	3	7	3	13	12	7	3	7	5	10	2	1	383
4	Callum Ilott	Ret	2	1	9	7	1	Ret	3	2	5	1	2	Ret	9	9	14	6	4	5	1	19	4	3	4	1	4	Ret	4	1	5	344
5	Jake Hughes	13	3	13	10	13	Ret	Ret	6	Ret	2	4	7	Ret	2	5	Ret	4	Ret	8	2	5	2	1	2	11	13	16	12	5	8	207
6	Jehan Daruvala	10	8	6	2	8	9	10	9	11	3	8	9	6	4	1	4	5	5	9	16	14	6	10	5	13	5	6	5	8	20	191
7	Ferdinand Habsburg	12	13	12	3	5	5	8	8	6	15	10	12	Ret	15	8	8	1	6	4	6	2	5	6	Ret	6	9	4	3	20	Ret	187
8	Guanyu Zhou	7	7	Ret	5	6	10	Ret	Ret	10	7	3	4	3	8	12	12	17	3	16	8	4	9	14	Ret	9	19	14	13	3	3	149
9	Ralf Aron	Ret	16	10	8	9	8	5	5	3	Ret	12	13	9	5	4	11	8	7	Ret	10	8	14	5	3	12	6	18	8	10	19	123
10	Nikita Mazepin	Ret	15	7	11	10	11	4	7	Ret	12	11	10	10	18	10	2	7	11	11	11	10	Ret	11	16	Ret	3	2	6	6	7	108
11	Harrison Newey	6	10	9	17	Ret	15	6	4	4	6	15	18	5	7	15	7	13	9	10	4	7	12	8	6	19	12	11	14	14	17	106
12	Mick Schumacher	8	6	17	6	3	6	9	11	12	9	9	11	7	12	Ret	6	9	8	6	9	11	8	15	11	7	10	8	11	18	18	94
13	Joey Mawson	5	11	8	14	16	Ret	Ret	16	Ret	4	7	8	Ret	13	11	5	10	10	14	14	9	3	7	20	18	8	7	15	15	9	83
14	Pedro Piquet	9	18	11	16	Ret	7	7	13	8	17	13	14	2	6	Ret	10	16	12	7	Ret	6	Ret	17	13	10	15	15	7	7	6	80
15	Tadasuke Makino	11	19	15	12	14	13	13	12	7	13	17	17	8	14	Ret				Ret	15	16	7	4	19	5	17	3	9	9	11	57
16	David Beckmann	14	17	14	Ret	15	14	11	14	Ret	11	5	5	12	11	6	Ret	15	16	Ret	5	13	Ret	16	15	8	11	9	17	13	10	45
17	Jake Dennis	2	5	5	Ret	Ret	Ret	Ret	10	9																						41
18	Max Defourny																						11	18	9							2
19	Marino Sato	16	12	18	13	11	12	14	Ret	13	16	16	15	13	16	13	15	11	14	15	18	17	16	Ret	12	14	16	10	18	19	16	1
20	Sacha Fenestraz																						15	13	10							1
21	Keyvan Andres	15	14	16	15	12	Ret	12	15	Ret	14	18	16	14	17	14	13	12	13	12	13	15	17	19	14	15	14	Ret	20	Ret	14	0
22	Ameya Vaidyanathan																16	14	15	13	17	18	19	21	18							0
23	Petru Florescu																						18	20	17	17	18	13				0
	Guest drivers ineligible to score points																															
	Jüri Vips																												21	12	12	0
	Devlin DeFrancesco																									16	20	12	19	17	14	0
	Felipe Drugovich																												16	16	15	0
Pos.	Driver	SIL			MNZ			PAU			HUN			NOR			SPA			ZAN			NÜR			RBR			HOC			Points

Figure 3: Race results data of the 2017 FIA F3 European Championship.
Source: `https://en.wikipedia.org/wiki/2017_FIA_Formula_3_European_Championship`

Team	Chassis	Engine	No.	Driver	Status	Rounds
Motopark	F315/007	Volkswagen	1	Joel Eriksson		All
	F315/003		10	Petru Florescu		8–9
			4	Jüri Vips	G	10
	F314/016		33	Marino Sato	R	All
	F317/010		47	Keyvan Andres		All
	F316/020		55	David Beckmann		4–10
Prema Powerteam	F316/016	Mercedes	3	Maximilian Günther		All
	F315/004		8	Guanyu Zhou		All
	F317/003		25	Mick Schumacher	R	All
	F314/015		53	Callum Ilott[17]		All
Van Amersfoort Racing	F316/017	Mercedes	5	Pedro Piquet		All
	F312/051		17	Harrison Newey		All
	F317/005		55	David Beckmann		1–3
			15	Max Defourny	R	8
			16	Felipe Drugovich	G	10
	F316/010		96	Joey Mawson	R	All
Hitech Grand Prix	F316/019	Mercedes	7	Ralf Aron		All
	F317/011		11	Tadasuke Makino		1–5, 7–10
	F315/010		34	Jake Hughes		All
	F316/018		99	Nikita Mazepin		All
Carlin	F316/014	Volkswagen	21	Jake Dennis		1–3
	F312/010			Ameya Vaidyanathan		6–8
	F312/010			Devlin DeFrancesco	G	9–10
	F315/001		27	Jehan Daruvala	R	All
	F317/001		31	Lando Norris	R	All
	F312/004		37	Sacha Fenestraz	R	8
	F316/014		62	Ferdinand Habsburg		All

Figure 4: Driver - team combinations of the 2017 FIA F3 European Championship. Source: `https://en.wikipedia.org/wiki/2017_FIA_Formula_3_European_Championship`

3.2 Data preparation

The data as collected from Wikipedia and driverdb.com are transformed into a data set that contains the race results of all drivers in all championships that we consider. Each row represents a driver's race result in a particular championship year. A description of the columns is given in Table 1.

Column	Description
Driver	The driver's name.
Position	The finishing position in the particular race.
Race	An abbreviation of the name of the race. For example, 'ZAN' corresponds to a race at the Zandvoort circuit. When multiple races were held on the same circuit, the abbreviation also contains a number.
Team	The name of the team the driver was racing for.
Championship	The name of the championship in which the result was achieved.
Year	The year (season) in which the result was achieved.

Table 1: Description of the columns of the race results data set.

We can determine how many championship points a driver has scored in each race by using the so-called points system. The points system describes how many championship points are assigned to each finishing position. However, the points system differs per championship because the number of scoring drivers and, as a result, the number of championship points assigned to a position differ per championship. Moreover, some championships award the driver on pole position or the driver that drove the fastest lap with one point. For example, the All-Japan Formula Three Championship assigns points to the top 6 drivers based on a 10-7-5-3-2-1 points system and awards the pole position and the fastest lap with one extra point each, while the Brasil F3 Championship assigns points to the top 8 drivers based on a 15-12-9-7-5-3-2-1 points system with no extra points for the pole position or the fastest lap.
We introduce a generic points system based on the points system that is used in the FIA F3 European Championship. This system assigns points to the top 10 drivers as 25-18-15-12-10-8-6-4-2-1. Additionally, we adapt the fractional points system as introduced by Phillips (2014). Finishing positions below 10th place are awarded with fractional points, computed using the formula

$$\text{number of points pth position} = a^{p-10}. \tag{4}$$

We choose $a = \left(\frac{1}{25}\right)^{\frac{1}{9}}$ to reflect the change from 25 points for first place to 1 points for tenth place in the points calculation. Table 2 contains the adjusted points system.

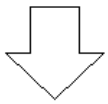
Position	**Points**
1	25 points
2	18 points
3	15 points
4	12 points
5	10 points
6	8 points
7	6 points
8	4 points
9	2 points
10	1 point
p $>$ 10	a^{p-10} points, where $a = \left(\frac{1}{25}\right)^{\frac{1}{9}}$

Table 2: Adjusted points system.

This adjusted points system can only be used for scoring and non-scoring finishes but we also have to handle the variety of non-finishes that can occur. We do not distinguish between non-finishes caused by a driver or a car failure because the data set does not contain information about the cause of a retirement. All non-finishes are treated as if the driver did not compete in that race and assigned zero championship points. In a certain sense, we thus adopt the way in which Bell et al. (2016) handle non-finishes.

We aggregate the race results to obtain a new data set that summarizes the race results for each driver - team combination in each season, as illustrated in Figure 5. This data set contains variables that describe the distribution of championship points and finishing positions for each combination of driver, team, championship, and year. Table 3 gives an overview of the variables that are included in this data set. These variables will be used to create the general performance features that describe the performance during a driver's career path to F3. For the karting championships, we cannot define variables that describe the distribution of the championship points or the finishing positions since we only have championship standings rather than race results data. Table 4 gives an overview of the variables that are used for kart championships.

Position	Race	Championship	Year	Points	Team
1	SIL	FIA Formula 3 European Championship	2017	25.00000000	Carlin
9	SIL.1	FIA Formula 3 European Championship	2017	2.00000000	Carlin
3	SIL.2	FIA Formula 3 European Championship	2017	15.00000000	Carlin
1	MNZ	FIA Formula 3 European Championship	2017	25.00000000	Carlin
2	MNZ.1	FIA Formula 3 European Championship	2017	18.00000000	Carlin
2	MNZ.2	FIA Formula 3 European Championship	2017	18.00000000	Carlin
2	PAU	FIA Formula 3 European Championship	2017	18.00000000	Carlin
2	PAU.1	FIA Formula 3 European Championship	2017	18.00000000	Carlin
Ret	PAU.2	FIA Formula 3 European Championship	2017	0.00000000	Carlin
8	HUN	FIA Formula 3 European Championship	2017	4.00000000	Carlin
14	HUN.1	FIA Formula 3 European Championship	2017	0.23916263	Carlin
3	HUN.2	FIA Formula 3 European Championship	2017	15.00000000	Carlin
11	NOR	FIA Formula 3 European Championship	2017	0.69931579	Carlin
1	NOR.1	FIA Formula 3 European Championship	2017	25.00000000	Carlin
3	NOR.2	FIA Formula 3 European Championship	2017	15.00000000	Carlin
1	SPA	FIA Formula 3 European Championship	2017	25.00000000	Carlin
Ret	SPA.1	FIA Formula 3 European Championship	2017	0.00000000	Carlin
1	SPA.2	FIA Formula 3 European Championship	2017	25.00000000	Carlin
1	ZAN	FIA Formula 3 European Championship	2017	25.00000000	Carlin
3	ZAN.1	FIA Formula 3 European Championship	2017	15.00000000	Carlin
1	ZAN.2	FIA Formula 3 European Championship	2017	25.00000000	Carlin
1	NÜR	FIA Formula 3 European Championship	2017	25.00000000	Carlin
2	NÜR.1	FIA Formula 3 European Championship	2017	18.00000000	Carlin
1	NÜR.2	FIA Formula 3 European Championship	2017	25.00000000	Carlin
4	RBR	FIA Formula 3 European Championship	2017	12.00000000	Carlin
2	RBR.1	FIA Formula 3 European Championship	2017	18.00000000	Carlin
17	RBR.2	FIA Formula 3 European Championship	2017	0.08179247	Carlin
2	HOC	FIA Formula 3 European Championship	2017	18.00000000	Carlin
11	HOC.1	FIA Formula 3 European Championship	2017	0.69931579	Carlin
4	HOC.2	FIA Formula 3 European Championship	2017	12.00000000	Carlin



Driver	Championship	Team	Year	position.champ	mean.points	mean.position	...
Lando Norris	FIA Formula 3 European Championship	Carlin	2017	1	14.75732	4.035714	...

Figure 5: Transformation of the race results data set; all race results of a driver-team combination in a certain season of a championship are aggregated.

Variable	Description
Driver	The driver's name.
Team	The name of the team the driver was racing for.
Championship	The name of the championship in which the race results were achieved.
Year	The year (season) in which the race results were achieved.
Champ.position	Position in the driver championship.
Mean.points	Mean of the number of championship points scored by the driver.
Mean.position	Mean of the driver's finishing positions.
Sd.position	Standard deviation of the driver's finishing positions.
Median.points	Median of the number of championship points scored by the driver.
Median.position	Median of the driver's finishing positions.
Quantile25.points	25% quantile of the number of championship points scored by the driver.
Quantile75.points	75% quantile of the number of championship points scored by the driver.
Quantile25.position	25% quantile of the driver's finishing positions.
Quantile75.position	75% quantile of the driver's finishing positions.
Min.points	Minimum of the number of championship points scored by the driver.
Min.position	Minimum of the driver's finishing positions.
Max.points	Maximum of the number of championship points scored by the driver.
Max.position	Maximum of the driver's finishing positions.
Perc.races	Percentage of races participated by the driver.
Perc.finished	Percentage of races finished by the driver.
First.places	Percentage of first places.
Podium.places	Percentage of podium places.
Toptwentypercent	Percentage of finishes in best 20 % positions (rounded above).
Topthirtypercent	Percentage of finishes in best 30 % positions (rounded above).
Topfourtypercent	Percentage of finishes in best 40 % positions (rounded above).
Frac.points	Fraction of the number of points scored by the champion.
Mean.rel.position	Mean of the relative position with respect to the best teammate.
Rel.points	Difference between the fraction of the championship points achieved by the driver and his/her best teammate.
Count	A driver's xth season in that championship.

Table 3: Performance variables single-seater auto racing championships.

Variable	Description
Driver	The driver's name.
Team	The name of the team the driver was racing for.
Championship	The name of the championship in which the race results were achieved.
Year	The year (season) in which the race results were achieved.
Champ.points	Mean of the number of championship points scored by the driver.
Champ.position	The driver's championship position.
Frac.points	Fraction of the number of points scored by the champion.
Count	A driver's xth season in that championship.
Rel.points	Difference between the fraction of the championship points achieved by the driver and his/her best teammate.

Table 4: Performance variables kart championships.

We now discuss how the mean relative position (*mean.rel.position*) and the relative points (*rel.points*) are computed. The other variables are straightforward to compute.

Mean relative position (mean.rel.position)
The mean relative position can be used to describe how good a driver performs relative to his/her teammate(s). This can be a good measure of driver performance as all drivers within a team have similar cars. The relative position per race is computed by dividing the position of the driver by the position of the best teammate. The best driver of the team receives thus a score of 1, while the other driver receives a score bigger than 1 for which holds that the closer to 1 the better. When a driver and/or none of drivers in the team finish the race, the relative position will be set to *NA*. The relative position is also set to *NA* if the driver does not have any teammates because otherwise the driver unfairly receives the score of 1 while he/she did not beat any teammates. The driver's mean relative position can be computed by taking the mean over the relative positions in all races during a season.

Relative points (rel.points)
The relative points is another measure that describes how good a driver performs relative to his/her teammate based on the difference in championship points scored. The relative points are given by the difference between the fraction of points (frac.points) scored by the driver and the fraction of points scored by his/her best teammate. The relative points are set to *NA* if the driver does not have any teammates.

3.3 Data cleaning

Finally, the race results data set is cleaned by removing inconsistencies in driver names and team names. Most of these inconsistencies need to be solved manually, but computing the approximate distance between pairs of strings can speed up this process. The *adist* function in R is used to compute the approximate distance between each pair of driver names in the data set. The computed distance is a generalized Levenshtein distance, giving the minimal possibly weighted number of insertions, deletions, and substitutions needed to transform one string into another. Then, all pairs of driver names are retrieved that have a Levenshtein distance no greater than three and inconsistencies are solved manually.

4 Feature engineering

This section describes the feature engineering process. We create features that describe a driver's performance during his/her career in single-seater auto racing and karting. The race results data set is used to create three different types of features:

1. General performance features
   General performance features describe a driver's performance during his/her career in single-seater auto racing and karting but are **not** corrected for team and competition effects. These features can be directly derived from the race results data and include statistics that describe the distribution of finishing positions or championship points within a certain championship. Moreover, some of the general performance features describe a driver's performance relative to his teammate(s). All performance variables from Table 3 are transformed into general performance features. We create general performance features for each season of each championship that the driver has competed in.

2. Driver performance features
   Driver performance features also describe a driver's performance during his/her career in single-seater auto racing and karting. They are considered to be a more reliable measure of driver performance than the general performance features as they are corrected for team and competition effects. We estimate a non-linear model that can distinguish between driver performance, team performance, and competition effects. The results of this model can be used to compute an adjusted points rate, that is, the mean number of championship points per race corrected for team performance and competition effects. The adjusted points rate is then added to the data set as driver performance feature. We create such a feature for each season of each championship that the driver has competed in, provided that there are a sufficient number of observations available for that particular championship to estimate the non-linear model. Section 4.3 describes in more detail how the driver performance features are created.

3. Driver features
   Driver features include some driver characteristics that we consider to be important for a driver's career in F3, such as age, experience (in single-seater auto racing and karting), nationality and team.

We remove observations belonging to drivers that competed in less than 20% of the races because we believe that these observations do not accurately reflect the driver performance in a season. The response variable should describe the driver performance in the FIA F3 European Championship (2008-2017). We choose the adjusted points rate in the FIA F3 European Championship (2008-2017) as the response variable since we believe that this is a more reliable measure of driver performance than, for example, the points rate itself. We will now discuss in more detail how the different types of features are created.

4.1 General performance features

General performance features describe the performance in relevant single-seater auto racing championships and karting championships, but do not distinguish between driver performance and team/car performance. As a result, these features can be biased when they are used to measure driver performance. For example, a driver that wins all races and has the best car is not necessarily the most talented driver.

We use all performance variables from Table 3 to create the general performance features. When a driver competed for multiple seasons in the same championship, we create general performance features for each of these seasons. For example, Mick Schumacher competed for 2 seasons in the ADAC Formula 4 Championship. He ended 10th in 2015 and 2nd in 2016. Then we create the performance features for

his first as well as his second season in the ADAC Formula 4 Championship. This is illustrated in Table 5 using the performance variable *champ.position*, which is Mick Schumacher's position in the driver championship. However, in the actual data set the same strategy is used for all general performance features, which would result in 46 features describing Mick Schumacher's performance in the ADAC Formula 4 Championship over 2 seasons.

Driver	**(other features)**	**ADAC F4 (1)**	**ADAC F4 (2)**	**Performance F3**
Mick Schumacher	...	10	2	

Table 5: Mick Schumacher's career path in the ADAC F4 championship represented in the machine learning data set.

By doing this, we can distinguish a driver's rookie season from other seasons in a certain championship. In general, we can state that a driver performs worse in his rookie year compared to his second or third year because then he/she is more experienced, which is also the case for Mick Schumacher in the ADAC F4 Championship.

Most of the single-seater auto racing championships that we consider relevant for predicting the driver performance in F3 are lower championships than the FIA F3 European Championship. That means that these championships often occur in a young driver's career path to F3. However, the GP3 Series championship is similar to the FIA F3 European Championship and young drivers often choose between the FIA F3 European Championship and the GP3 Series when they completed the lower championships. It is even very likely that the FIA F3 European Championship and the GP3 Series are merged in 2019 to create one support series for F2 and F1.
Many drivers even compete in both the FIA F3 European Championship and the GP3 Series during different seasons. When a driver competes in the GP3 Series after his career in the FIA F3 European Championship it might be questionable whether these data could be added to a driver's career path. However, we decided to add future performance in the GP3 Series to the career path and not to distinguish between performance in the GP3 Series before and after competing in the FIA F3 European Championship. The most important argument is that we believe that the performance in the GP3 Series will a good predictor for the performance in the FIA F3 European Championship. As a result, we expect the explanatory power obtained by including future performance GP3 Series into the machine learning models to outweigh any inaccuracy that is caused by including future performance instead of past performance.

When a driver competed for multiple seasons in the FIA F3 European Championship, we can also use data from previous seasons as features for the machine learning model. For example, Antonio Giovinazzi competed for three seasons in the FIA F3 European Championship, as can be seen in Table 6.

Year	**Championship**	**Result**
2010	CIK FIA European Championship KF	18
2011	CIK FIA European Championship KF	5
2013	British Formula Three Championship	2
2013	FIA Formula 3 European Championship	17
2014	FIA Formula 3 European Championship	6
2015	FIA Formula 3 European Championship	2

Table 6: Antonio Giovinazzi's career path up to the FIA F3 European Championship.

Table 7 illustrates how Antonio Giovinazzi's F3 career is represented in the data set. In this example, we describe the performance in a championship using the final position in the championship standings, as opposed to the real data set where a variety of measures is used to describe driver performance. Also, for simplicity, we use the position in the championship as response variable while we use the adjusted points

rate in the real data set.
For each season that Antonio Giovinazzi has competed in the FIA F3 European Championship, a new row is created in the data set. The first row corresponds to his first season in F3 and contains features describing his performance in other relevant single-seater auto racing championships and karting championships. The second row corresponds to his second season in F3 and uses the performance during the first season as an additional feature. The third and last row corresponds to his most recent season in F3 and uses the performance during the previous two seasons as additional features.

Driver	Year	(other features)	Position 1	Position 2	Position 3	Y
Antonio Giovinazzi	2013	...				17
Antonio Giovinazzi	2014	...	17			6
Antonio Giovinazzi	2015	...	17	6		2

Table 7: Antonio Giovinazzi's career path in the FIA F3 European Championship represented in the machine learning data set.

4.2 Driver features

Driver features include some driver characteristics that we consider to be important for a driver's performance in the FIA F3 European Championship. For example, one might expect that an older driver will perform better because he/she is more experienced. The driver features include age, nationality, and experience in single-seater auto racing and karting. Moreover, we include the team in which the driver participated during a F3 season. Table 8 gives a description of each of these features.

Driver feature	Description
team	The driver's team during that F3 season.
age	The driver's age during that F3 season.
nationality	The driver's nationality.
kart.experience	Number of years experienced in karting (all relevant categories).
race.experience	Number of years experienced in racing (all relevant categories).

Table 8: Driver features.

Note that we do not include the driver's current age but his/her age at the time of the F3 season. A driver's age in a certain season is thus computed as the difference between his/her year of birth and the year of the season. As a result, the age that we compute might differ one year from the actual age. Also, we only include nationalities that occur five times or more in the data set, otherwise we set the nationality to 'other'. When computing the experience in single-seater auto racing and karting we only consider the championships that occur in the data set. Other categories are not considered as relevant experience.

4.3 Driver performance features

As stated before, finishing positions and number of championship points scored are considered as unreliable measures of driver performance because they are influenced by team, car and competition effects. The general performance features are thus insufficient to describe a driver's performance during his career path. Therefore, we want to create features that describe the driver performance in a championship and are not influenced by team, car and competition effects. The statistical model as proposed by (Phillips, 2014) can be used to obtain rankings based on driver performance for the FIA F3 European Championship and other relevant championships of single-seater auto racing. This method provides us with an estimate of the driver performance and an adjusted average number of championship points scored solely

based on driver performance. Both measures can be used as features in the machine learning model to describe the performance along a driver's career path. However, Phillip's method cannot be applied to the kart championships because results per race and data about the teams are missing for most of the kart championships. Because of lack of data Phillip's method can also not be applied to some of the single-seater auto racing championships. We will now discuss the statistic model as proposed by (Phillips, 2014) in more detail, following Phillips' discussion closely.

4.3.1 Statistical model

The statistical model describes the influence of driver performance, team performance, and competition effects on the race results. We estimate the model for each single-seater auto racing championship l separately to obtain driver performance estimates for each of these championships.

We define the following variables for each driver i in each team j in each season k of championship l:

$$\begin{aligned} s_{ijkl} &= \text{ average scoring rate for driver } i \text{ in team } j \text{ in season } k \text{ of championship } l \\ &= \frac{\text{total number of championship points}}{\text{total number of races}} \\ y_{ijkl} &= \text{ underlying performance for driver } i \text{ in team } j \text{ in season } k \text{ of championship } l \end{aligned}$$

where $i = 1, \ldots, N$ with N the total number of drivers, $j = 1, \ldots, T$ with T the total number of teams and $k = 1, \ldots, K$ with K the total number of seasons. Note that a driver may compete for more than one team in one season, with a different s_{ijkl} for each team.

We assume that the performance variable y_{ijkl} is predictive of average scoring rate s_{ijkl}. In particular, we assume that s_{ijkl} is a sigmoidal function of y_{ijkl}

$$s_{ijkl} = S(y_{ijkl}) = \frac{25}{2}\left(1 + \operatorname{erf}(y_{ijkl})\right),$$

where S is a sigmoidal function and $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the standard error function. The sigmoidal function is chosen to be ranging from 0 to 25 because the minimum number and maximum number of points that can be scored in a race are equal to 0 and 25, respectively, according to the adjusted points system.

For each s_{ijkl}, the value of y_{ijkl} can be computed by applying the inverse function

$$y_{ijkl} = S^{-1}(s_{ijkl}) = \operatorname{erf}^{-1}\left(\frac{s_{ijkl} - \frac{25}{2}}{\frac{25}{2}}\right).$$

We assume that y_{ijkl} is a linear function of driver, team, and competition effects on performance

$$y_{ijkl} = \alpha_{il} + \left(1 - \frac{1}{n_{jl}}\right)\beta_{jl} - \delta_{kl} + \epsilon_{ijkl}, \tag{5}$$

where

$$\begin{aligned} \alpha_{il} &= \text{fixed effect representing performance driver } i \text{ in championship } l, \\ n_{jl} &= \text{number of observations in championship } l \text{ belonging to team } j, \\ \beta_{jl} &= \text{fixed effect representing performance team } j \text{ in championship } l, \\ \delta_{kl} &= \text{fixed effect representing the difficulty of scoring points in season } k \text{ of championship } l \\ &\quad \text{due to competition with other drivers ,} \\ \epsilon_{ijkl} &= \text{random effect representing variability in performance.} \end{aligned}$$

The model's random effects, ϵ_{ijkl}, are assumed to be independent normally distributed with mean zero and variance σ^2. For each driver i in each season k of championship l, we define a weighted predictor by averaging across all teams j the driver participated in

$$y_{ikl} = \frac{\sum_j w_{ijkl} y_{ijkl}}{\sum_j w_{ijkl}},$$

where w_{ijkl} is the number of races in which the driver has participated for that particular team.

We adjusted Phillip's model by introducing a weight function $1 - \frac{1}{n_{jl}}$ before the coefficient βjl in (5). Introducing this weight function influences the results in the following way. Teams that have more observations in a certain championship will be assigned a higher weight than teams that have fewer observations. This is desirable since we consider a coefficient more reliable when it is estimated using more data. Also, driver performance is often more important than team performance in championships lower than F1. While the driver performance and team performance were equally weighted in Phillip's model, it is more convenient that team performance is less important than driver performance in our model. Note that when $n_{jl} \to \infty$ driver performance and team performance are equally important. However, for most of the teams, only a limited number of observations is present in the data set resulting in a weight of the team coefficient smaller than 1 and a team coefficient that is less important than the driver coefficient.

4.3.2 Competition effects

Since there is only a limited number of championship points that can be earned in each race, competition with other drivers makes it more difficult to score points. The level of competition is expected to vary from season to season, depending on the number of other drivers and teams and their performance. The effect of the level of competition is modeled by including the term δ_{kl} in (5).

We can estimate the form of δ_{kl} by considering the expected scoring rate of N drivers. We assume that their performances $\gamma_{ijl} = \alpha_{il} + \beta_{jl}$ are normally distributed with mean μ and standard deviation σ. The expected total scoring rate for N drivers can be computed as

$$\begin{aligned}
\mathbb{E}(s_{\text{tot}}) &= N\,\mathbb{E}(s_{ijkl}) \\
&= N \int_{-\infty}^{\infty} S(y)p(y)dy \\
&= N \int_{-\infty}^{\infty} \frac{25}{2}(1+\operatorname{erf}(y))\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(y+\delta-\mu)^2}{2\sigma^2}} dy \\
&= \frac{25}{2} N \left[\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(y+\delta-\mu)^2}{2\sigma^2}} dy + \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \operatorname{erf}(y-\delta-\mu) e^{-\frac{y^2}{2\sigma^2}} dy \right] \\
&= \frac{25}{2} N \left[1 + \operatorname{erf}\left((-\delta+\mu)\sqrt{\frac{1}{1+2\sigma^2}} \right) \right].
\end{aligned}$$

Then we can use the fact that the total scoring rate s_{tot} must be equal to the total number of points on offer per races p_{tot}. p_{tot} can be computed as

$$p_{tot} = 25+18+15+12+10+8+6+4+2+1+\sum_{p=11}^{N} a^{p-10}, \text{ with } a = \left(\frac{1}{25}\right)^{(1/9)}.$$

Rearranging the terms in (13) gives us an expression for the competition effect δ,

$$\delta = \mu - \sqrt{1+2\sigma^2}\,\operatorname{erf}^{-1}\left(\frac{P_{\text{tot}} - \frac{25}{2}N}{\frac{25}{2}N} \right).$$

The competition function

$$\delta_{kl}(\mu_{kl}, \sigma_{kl}, N_{kl}) = \mu_{kl} - \sqrt{1 + 2\sigma_{kl}^2}\, \text{erf}^{-1}\left(\frac{P_{\text{tot}} - \frac{25}{2}N_{kl}}{\frac{25}{2}N_{kl}}\right),$$

describes the level of competition of season k in championship l. The values of m_{kl}, σ_{kl}, and N_{kl} are computed by weighting each driver/team computation by w_{mnkl}/R_{kl}, where w_{mnkl} is the number of races completed by that driver/team combination in season k of championship l and R_k is the total number of races in season k of championship l. Thus, the parameters of the competition function can be computed as

$$N_{kl} = \sum_{m,n} \frac{W_{mnkl}}{R_{kl}},$$

$$\mu_{kl} = \sum_{m,n} \frac{w_{mnkl}\theta_{mnkl}}{R_{kl}N_{kl}}, \text{ and}$$

$$\sigma_{kl}^2 = \sum_{m,n} \frac{w_{mnkl}(\theta_{mnkl} - \mu_{kl})^2}{R_{kl}N_{kl}}.$$

4.3.3 Model fitting

The model's random effects, ϵ_{ijk}, are assumed to be independent and normally distributed with mean zero and variance σ^2. The model's maximum likelihood is thus achieved by estimating the values of $\hat{\alpha}_{il}$ and $\hat{\beta}_{jl}$ that correspond to the least-squares fit for y_{ikl} to the data. That is, we want to minimize the sum of squared errors given by

$$SSE(\hat{\theta}) = \sum_{i=1}^{n}(y_{ijkl} - \hat{y}_{ijkl})^2$$

with respect to the parameter vector θ, which consists of the α_{il}'s and β_{il}'s. Here n represents the total number of observations. The parameter vector θ is estimated using the *nls.lm* function from the *minpack.lm* package in R (Elzhov, Mullen, Spiess, & Bolker, 2016). The *nls.lm* function provides an R interface to the *lmder* and *lmdif* functions from the *MINPACK* library, for solving nonlinear least-squares problems by a modification of the Levenberg Marquardt algorithm.

Drivers that score 0 points or 25 points in every race of a season cause potential problems for fitting, since these scores theoretically correspond to $y_{ijkl} \to \infty$ and $y_{ijkl} \to -\infty$, respectively. We avoid these potential problems by conservatively estimating the scoring rate, i.e., by assuming that with enough races, a driver would eventually stop scoring 0 or 25 points, respectively.

- When a driver scores 25 points in every race, we conservatively estimate the scoring rate by supposing the existence of one additional race in which the driver finishes second (scored 18 points), such that

  $$y_{ijkl} = \frac{25w_{ijkl} + 18}{w_{ijkl} + 1}.$$

  This method allows us to differentiate between drivers with perfect records by giving a higher scoring rate to drivers who achieved perfect records over more races. This approach is justified by noting that it rewards drivers for more statistically reliable perfect performances, e.g., 10 wins in 10 races is more impressive than one win in one race.

- When a driver scores 0 points in every race, we conservatively estimate the scoring rate by supposing the existence of one additional race in which the driver finished the lowest position that any driver finished in that season, such that

  $$y_{ijkl} = \frac{25w_{ijkl} + \text{min.points}}{w_{ijkl} + 1},$$

where min.points is the number of championship points associated with the lowest finishing position in that season. Again, we differentiate between 'bad' drivers because we give a lower scoring rate to drivers that score no points over more races.

During the fitting process, it also appeared that the data from championships we consider are not as suitable to fit this model to as F1 data. This is probably because of the following reasons:

- Phillips fits the model on a large data set using F1 race results data from 1950-2013, while we fit the model for each championship separately on at most nine years of data. However, it is not desirable to use more data because we need to reflect the common career path an F3 driver takes in the data. We believe that data from before 2008 is not representative for this end.
- While teams often compete for a long period of time in F1, this is not necessarily the case for lower championships. It is easier to step into a lower championship than it is to step into F1. Some teams also compete in different championships and every year they choose again in which championships they will compete.
- While drivers often compete for a long period of time in F1, this is definitely not the case for lower championships. Lower championships are mostly used to educate young drivers, so drivers often compete only for one or two seasons in the same championship.

These differences between F1 and other championships, such as F3, make it more difficult to estimate Phillip's model on our data set. Some team coefficients cannot be estimated in a reliable way because of lack of observations. Therefore, observations of teams that only occur in combination with three or fewer drivers are removed from the data set. When a team only occurs in combination with a small number of unique drivers the model will be unable to distinguish the driver effect from the team effect and, as a result, the driver and team coefficients will be incorrectly estimated. Also, observations of teams that only compete in one season are removed from the data set because in that case there is also not enough information available in the data to obtain reliable estimates of team performance. However, even without these teams, the team coefficients can highly fluctuate from one to another, which makes it harder to compute the adjusted points rates. To obtain usable adjusted points rates, we adjust the weighted team coefficients by removing the 'outliers'. We define an outlier as a team coefficient that lies outside the range of 1.5 times the inter-quantile distance. These team coefficients are adjusted to be equal to 25% quantile or 75% quantile depending on whether they are negative or positive outliers. Finally, we apply a weighting function that assigns a score closer to the minimum score when there are fewer observations available. This weighting function is given by $\min_{\text{score}} + (1 - \alpha^n)(\text{score} - \min_{\text{score}})$, where $\alpha \in (0, 1)$. Since n is rather small, we choose a high α ($\alpha = 0.99$) to reflect the difference in number of observations in the team performances. Making these adjustments is not ideal because it affects the estimated team performances but it is needed to obtain usable adjusted points rates.

4.3.4 Adjusted scoring rates

The estimated parameter values can be used to compute the underlying driver performances. The underlying driver performance can be can be computed by correcting for team and competition effects

$$\tilde{y}_{ijkl} = y_{ijkl} - \tilde{\beta}_{jl} + \hat{\delta}_{kl} + \beta - \delta,$$

where $\tilde{\beta}_{jl}$ represents the adjusted team coefficient. The coefficients β and δ are baselines for the team and competition effects. Any baseline is arbitrary and will not affect the order of the rankings. Then we can compute adjusted scoring rates for each driver i in each season k of championship l by taking a weighted sum across all teams, j, for which they drove in that season,

$$\tilde{s}_{ikl} = S\left(\frac{\sum_j w_{jl}\, \tilde{y}_{ijkl}}{\sum_j w_{jl}}\right).$$

The adjusted scoring rate represents the average number of championship points that the driver scored in season k of championship l, purely based on driver performance. This gives us a more reliable measure of driver performance than the actual scoring rate, because the adjusted scoring rate is adjusted for team/car performance and competition effects.

4.3.5 Accuracy of the estimators

The accuracy of the estimated driver coefficients α_{il} and estimated team coefficients β_{jl} can be quantified using a confidence interval. We can only construct confidence intervals for the original team coefficients β_{jl} and not for the adjusted team coefficients $\tilde{\beta}_{jl}$, which we obtained by setting the outliers to the 25% or 75% quantiles and using the smoothing function. Confidence intervals for the competition effects δ_{kl} are not constructed, because the competition effects are a function of the parameters α_{il} and β_{jl}. Let θ be the vector consisting of all driver coefficients α_{il} and team coefficients β_{jl}, where $i = 1, \ldots, N$ and $j = 1, \ldots, T$. The estimator of the covariance matrix of $\hat{\theta}$ is given by

$$\hat{\text{Cov}}(\hat{\theta}) = \hat{\sigma}^2(\hat{V}^T\hat{V})^{-1}$$

where $\hat{V}$ is the Jacobian matrix evaluated at $\hat{\theta}$ and $\hat{\sigma}$ is an estimate for the standard deviation of the residuals e_{ijkl}. The matrix $\hat{V}$ is computed numerically using the function *nl.jacobian* of R package *nloptr* (Johnson, 2014). Moreover, $\hat{\sigma}^2$ is computed by

$$\hat{\sigma}^2 = \frac{R^T R}{n - p},$$

where R is the vector consisting of the residuals $\hat{e}_{ijkl}$, n is the total number of observations and p is the number of estimated parameters. Note that here p is equal to the number of drivers N plus the number of teams T. An approximate $(1 - \alpha)100$ % confidence interval for the θ_m, $m = 1, \ldots, p$, is defined as

$$\hat{\theta}_j \pm t_{(n-p);(1-\alpha/2)}\hat{\sigma}\sqrt{((\hat{V}^T\hat{V})^{-1})_{mm}}.$$

The inverse of the matrix $(\hat{V}^T\hat{V})$ is computed numerically using the function *ginv* from the R package *MASS* (Venables & Ripley, 2002), which computes the Moore-Penrose generalized inverse of a matrix. The Moore-Penrose generalized inverse of a matrix is a pseudoinverse A+ of a matrix A. We compute the pseudoinverse of $(\hat{V}^T\hat{V})$ to deal with the problem that this matrix is non-identifiable for most of the championships.

4.3.6 Results

Confidential

4.3.7 Creating the features

The results of the statistical model are used in the machine learning models in two ways, as feature and as response variable. We include the adjusted points rate in other relevant championship of single-seater auto racing as driver performance features in the data set. When a driver competed for multiple seasons in the same championship, we create driver performance features for each of these seasons separately.
The adjusted points rate in the FIA F3 European Championship is used as response variable in the machine learning model. When a driver competed for multiple seasons in the FIA F3 European Championship, we create an observation for each of these seasons along with the corresponding adjusted points rate, as discussed in Section 4.1. However, the adjusted points rate could not be estimated for all F3 drivers because observations belonging to teams with not enough data were removed from the data set. For these drivers, we use the mean number of championship points scored by the driver (*mean.points*) as response variable. As stated before, we could not estimate the statistical model for all single-seater auto racing championships because some championships do not have enough observations compared to the number of parameters.

5 Feature analysis

This section provides an exploratory data analysis of the data set containing the features for the machine learning models. Based on this analysis we can select the features that are expected to have the highest predictive power when included in the machine learning models.

5.1 Data set containing the features

The goal of this research is to predict the driver performance in the FIA F3 European Championship based on a driver's career path. The feature engineering process of Section 4 was focused on creating features that describe a driver's performance during his/her career in single-seater auto racing and karting. This has resulted in a data set containing 268 observations of 982 features in total. Each observation in the data set represents a driver's performance during an F3 season and his/her performance in other relevant championships of single-seater auto racing and karts. Drivers that competed for multiple seasons in the FIA F3 European Championship have an observation for each of these seasons, as was explained in Section 4. The data set contains data of 181 (previous) F3 drivers. Since we know the performance of these drivers in the FIA F3 European Championship, the machine learning models can try to learn the relationship between the performance in the FIA F3 European Championship and the performance in the past. Then we can predict the performance of potential F3 drivers, i.e., drivers that already compete in F3 and drivers of other relevant championships, in the FIA F3 European Championship.

We consider all F3 drivers from 2008-2017 that competed in at least 20% of the races. The number of competing drivers and teams differ per season, as can be seen in Figure 9. Note that the FIA F3 European Championship was called the Formula 3 Euro Series from 2008-2011.

Championship	**Year**	**# Drivers**	**# Teams**
Formula 3 Euro Series	2008	33	11
Formula 3 Euro Series	2009	33	10
Formula 3 Euro Series	2010	16	6
Formula 3 Euro Series	2011	15	8
FIA Formula 3 European Championship	2012	28	11
FIA Formula 3 European Championship	2013	33	12
FIA Formula 3 European Championship	2014	29	10
FIA Formula 3 European Championship	2015	35	10
FIA Formula 3 European Championship	2016	24	7
FIA Formula 3 European Championship	2017	22	5

Table 9: Number of drivers and teams per season.

5.2 Data analysis

We now provide an exploratory data analysis of the data set containing the features for the machine learning models. This exploratory data analysis consists of a descriptive analysis and computing correlations between the features and the response variable.

5.2.1 Descriptive analysis

Drivers can follow very different career paths before they end up in the FIA F3 European Championship. For example, Max Verstappen stepped into the FIA F3 European Championship right after his career in karts, while others first competed in other single-seater auto racing championships. As a result, the data points are unequally distributed over the different championships, as can be seen in Table 10. This table

shows the percentage of non-missing observations for each championship per season. From this table we can conclude that most drivers have competed in GP3 Series, Eurocup Formula Renault 2.0, British Formula 3 International Series, British Formula Three Championship and Formula Renault 2.0 NEC before stepping into the FIA F3 European Championship. Moreover, we have a relatively large number of data points from previous seasons in the FIA F3 European Championship. The CIK FIA European Championship KF and CIK FIA World Championship KF are the most important kart championships according to Table 10. Finally, we can conclude that the data set contains a large number of missing values, which is caused by the fact that not all drivers have competed in all different championships. The heterogeneous distribution of missing values among the variables as depicted in Table 10 is one of the biggest challenges of this data set. We choose not to impute the missing values with an imputation method because then the majority of the data set would consist of imputed data. Instead, we choose machine learning models that can handle missing values in the training data. These machine learning models along with the methods that they use to handle missing values will be discussed in Section 6.

Championship	**Season 1**	**Season 2**	**Season 3**	**Season 4**
FIA F3 European Championship	32.46	7.46	1.49	0.75
GP3 Series	27.24	8.58	0.75	
Eurocup Formula Renault 2.0	19.78	7.09	1.12	
British Formula 3 International Series	18.66	2.99		
British Formula Three Championship	17.54	1.49		
Formula Renault 2.0 NEC	14.93	6.34		
CIK FIA European Championship KF	12.31	3.36	0.37	
CIK FIA World Championship KF	7.84	0.37		
European F3 Open Championship	5.60	1.12		
Italian F4 Championship	5.22	0.37		
ADAC Formula 4	4.85	0.75		
BRDC Formula 4 Championship	3.73	0.37		
Euro Formula Open	3.73	1.12		
Formula Renault 2.0 WEC	3.36			
All Japan Formula Three Championship	2.61	0.75	0.37	
CIK FIA World Championship KZ	1.87			
Formula 3 Brasil	1.49	0.75		
MSA Formula Championship	1.12			
BRDC British Formula 3 Championship	0.75			
CIK FIA European Championship KZ	0.75			
F4 British Championship	0.75			
F4 Japanese Championship	0.75			
WSK Euro Series KZ1	0.75			
SMP F4 Championship	0.37			

Table 10: Percentage of non-missing values per season for each championship.

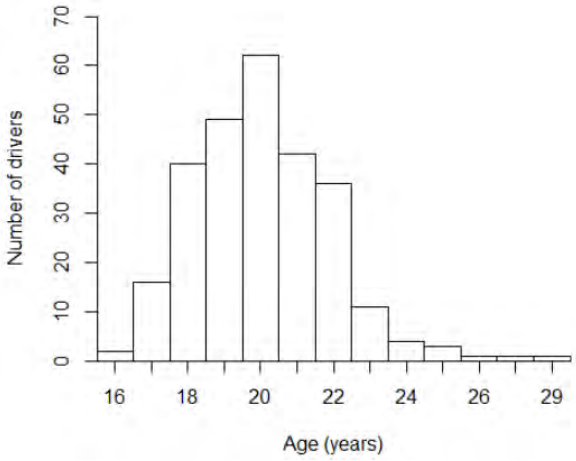
We now look at the career path of the past F3 champions in our data set, to give the reader some intuition on how such a career path might look like. Table 11 presents all F3 champions from 2008-2017 along with the relevant championships from their career path up to the season that they won the FIA F3 European Championship. This table shows that many past F3 champions already had some experience in the FIA F3 European Championship before winning this championship. Moreover, most champions competed in a Formula Renault championship, the British Formula 3 International Series and/or the British Formula Three Championship. We can thus conclude, based on Tables 10 and 11, that the past performance in those championships is expected to have high predictive power in the machine learning models.

Champion	Year	Career path
Lando Norris	2017	Karting
		ADAC Formula 4 (2015)
		Italian F4 Championship (2015)
		MSA Formula Championship (2015)
		BRDC British Formula 3 Championship (2016)
		Eurocup Formula Renault 2.0 (2016)
		Formula Renault 2.0 NEC (2016)
Lance Stroll	2016	Karting
		Italian F4 Championship (2014)
		FIA F3 European Championship (2015)
Felix Rosenqvist	2015	FIA F3 European Championship (2011)
		British Formula 3 International Series (2012)
		FIA F3 European Championship (2012)
		FIA F3 European Championship (2013)
		FIA F3 European Championship (2014)
Esteban Ocon	2014	Eurocup Formula Renault 2.0 (2012)
		Eurocup Formula Renault 2.0 (2013)
		Formula Renault 2.0 NEC (2013)
		GP3 Series (2015)
Rafaelle Marciello	2013	British Formula 3 International Series (2012)
		FIA F3 European Championship (2012)
Daniel Juncadella	2012	FIA F3 European Championship (2010)
		GP3 Series (2010)
		FIA F3 European Championship (2011)
		British Formula 3 International Series (2012)
Roberto Merhi	2011	Formula Renault 2.0 WEC (2008)
		FIA F3 European Championship (2009)
		FIA F3 European Championship (2010)
		GP3 Series (2010)
Edoardo Mortara	2010	FIA F3 European Championship (2008)
Jules Bianchi	2009	FIA F3 European Championship (2008)
		British Formula Three Championship (2009)
Nico Hülkenberg	2008	-

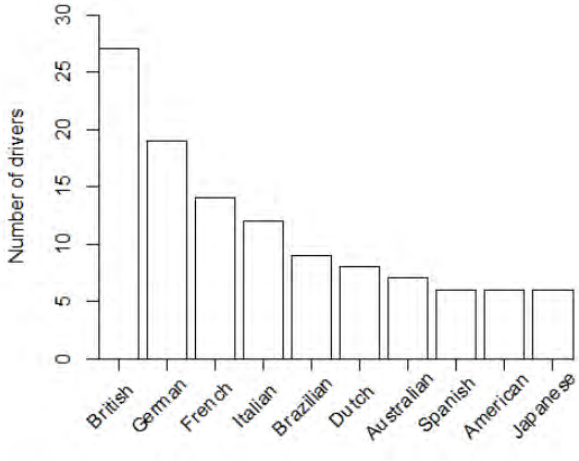
Table 11: Career path of past F3 champions.

Besides the performance in relevant championships of single-seater auto racing and karting, the data set also contains driver features that describe some characteristics of the F3 drivers. These driver features include age, race experience, kart experience and nationality. Figure 6a shows a barplot of the age distribution of F3 drivers at the time that they competed in the FIA F3 European Championship. The youngest driver was only 16 years old, while the oldest driver was 29 years old. Most F3 drivers were around 20 years old. Figure 6b shows a barplot of the nationality distribution of the drivers in the data set. From this plot we can conclude the most drivers were British, followed by German and French. We do not expect nationality to have high predictive power in the machine learning model since we believe that performance, age and experience are more important. However, some countries may have more opportunities for a young driver to become experienced in karting and/or auto racing. Finally, Figure 6c shows the teams the drivers were participating in. Most F3 drivers were participating in the Carlin team, which is not surprising since Carlin has competed in almost all seasons of the FIA F3 European Championship from 2008-2017. We will investigate whether the team influences the performance in the FIA F3 European Championship in the remainder of this report. However, this feature cannot be included in the final machine learning model since it is not known for new F3 drivers. This feature is also not expected

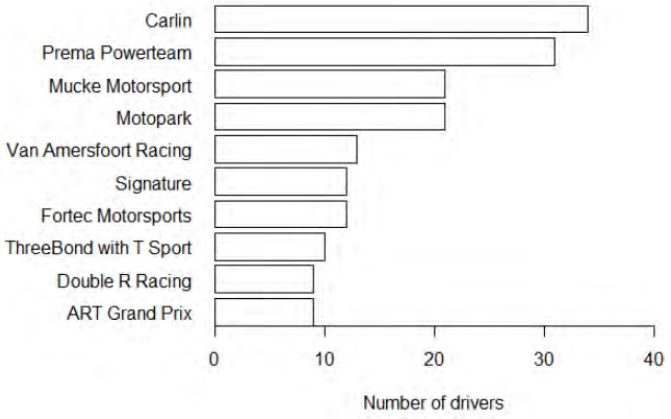
to have high predictive power since the response variable is corrected for team and competition effects.



(a) Barplot of the age among the F3 drivers.



(b) Barplot of the different nationalities among the F3 drivers.



(c) Barplot of the teams in which the F3 drivers participated.

5.2.2 Correlation analysis

We compute pairwise correlations between the features and the response variable to investigate which features are most likely to have predictive power in the machine learning models. We compute the Pearson correlation coefficient, which estimates linear correlation, as well as Spearman's correlation coefficient, which estimates both linear and non-linear correlation.

The Pearson correlation coefficient is a measure for linear correlation between two variables X and Y and can be computed as

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \in [-1, 1].$$

Here $\text{cov}(X,Y)$ denotes the covariance between the variables X and Y and σ_X and σ_Y denote the standard deviation of the variables X and Y, respectively. Its values lies between $+1$ and -1, where a value close to 1 means that there exists strong positive linear correlation between X and Y, a value close to

-1 means that there exists strong negative linear correlation between X and Y and, a value close to 0 means no linear correlation.

We also compute Spearman's rank correlation coefficient which measures the rank correlation between two variables X and Y, i.e., the statistical dependence between the ranks of X and Y. Spearman's correlation between two variables is equal to the Pearson correlation between the rank values of those two variables. While Pearson's correlation measures linear relationship, Spearman's correlation assesses monotonic relationships, both linear and non-linear. Let rg_{X_i} and rg_{Y_i} be the ranks of the variable X and Y, respectively. Spearman's rank correlation coefficient is then computed as the Pearson rank correlation coefficient between the rank variables:

$$r_s = \rho_{rg_{X_i} rg_{Y_i}} = \frac{\text{cov}(rg_{X_i}, rg_{Y_i})}{\sigma_{rg_{X_i}} \sigma_{rg_{Y_i}}} \in [-1, 1].$$

The interpretation of Spearman's rank correlation coefficient is similar to the interpretation of the Pearson correlation coefficient.

The Pearson and Spearman's correlation coefficients cannot be computed for missing values. Therefore, we only compute these coefficients for pairwise complete observations, i.e., observations that have no missing values in both the feature and the response variable. Also, we only compute the Pearson and Spearman's correlation coefficients for features that have at least ten pairwise complete observations with the response variable.

We expect the performance in previous seasons of the FIA F3 European Championship to be a good predictor for the performance in the next season. Moreover, we expect the performance in GP3 Series to be a good predictor for the performance in the FIA F3 European Championship, because these are similar championships. Table 12 contains the Pearson and Spearman's correlation coefficients computed between some features corresponding to the FIA F3 European Championship and GP3 Series and the response variable, respectively. Note that a strong negative correlation between any position feature and the response variable is desirable because positions and championship points have an opposite scale, i.e., the first position is awarded the most championship points. We can conclude that many of these performance features have indeed a correlation with the response variable higher than 0.5 in absolute value. However, we measured a relatively low correlation coefficient for the standard deviation of the position. This feature is thus not expected to have high predictive power when it is included in the machine learning models. Additional methods to measure the relevance of features of all championships will be discussed in Section 7.

	FIA European F3		**GP3 Series**	
Feature	**Pearson**	**Spearman**	**Pearson**	**Spearman**
Adjusted points rate (season 1)	0.667	0.710	0.552	0.562
Adjusted points rate (season 2)	0.686	0.673	0.771	0.751
Points rate (season 1)	0.652	0.699	0.545	0.555
Points rate (season 2)	0.661	0.603	0.675	0.666
Position championship (season 1)	-0.662	-0.720	-0.563	-0.591
Position championship (season 2)	-0.581	-0.472	-0.439	-0.495
Mean position (season 1)	-0.602	-0.656	-0.574	-0.578
Mean position (season 2)	-0.621	-0.618	-0.633	-0.641
Standard deviation position (season 1)	0.217	0.272	0.013	0.056
Standard deviation position (season 2)	0.078	0.109	-0.093	-0.055

Table 12: Pearson and Spearman's correlation coefficients computed between features corresponding to the FIA F3 European Championship and GP3 Series and the response variable.

One might expect that age and experience are good predictors for the performance in the FIA F3 European Championship. We investigate this by computing the correlation coefficients between age, race experience and kart experience and the response variable, respectively. The results are shown in Table 13. From this table we can conclude that correlation coefficients are rather low. The correlation between age and the response variable is close to 0, while race experience and kart experience show a slightly positive correlation with the response variable. This can also be concluded from the scatter plots in Figure 7, which do not show a clear linear relationship between the features age and experience and the response variable, respectively. Based on these results, the features age and experience are not expected to have high predictive power. However, it is possible that a more complicated relationship exists between these features and the response variable that is not reflected in the correlation coefficients. Additional methods to estimate feature importance are presented in Section 7.

Feature	**Pearson correlation**	**Spearman correlation**
Age	-0.001	-0.035
Race experience	0.115	0.123
Kart experience	0.131	0.114

Table 13: Pearson and Spearman correlation coefficients computed between the (numerical) driver features and the response variable.

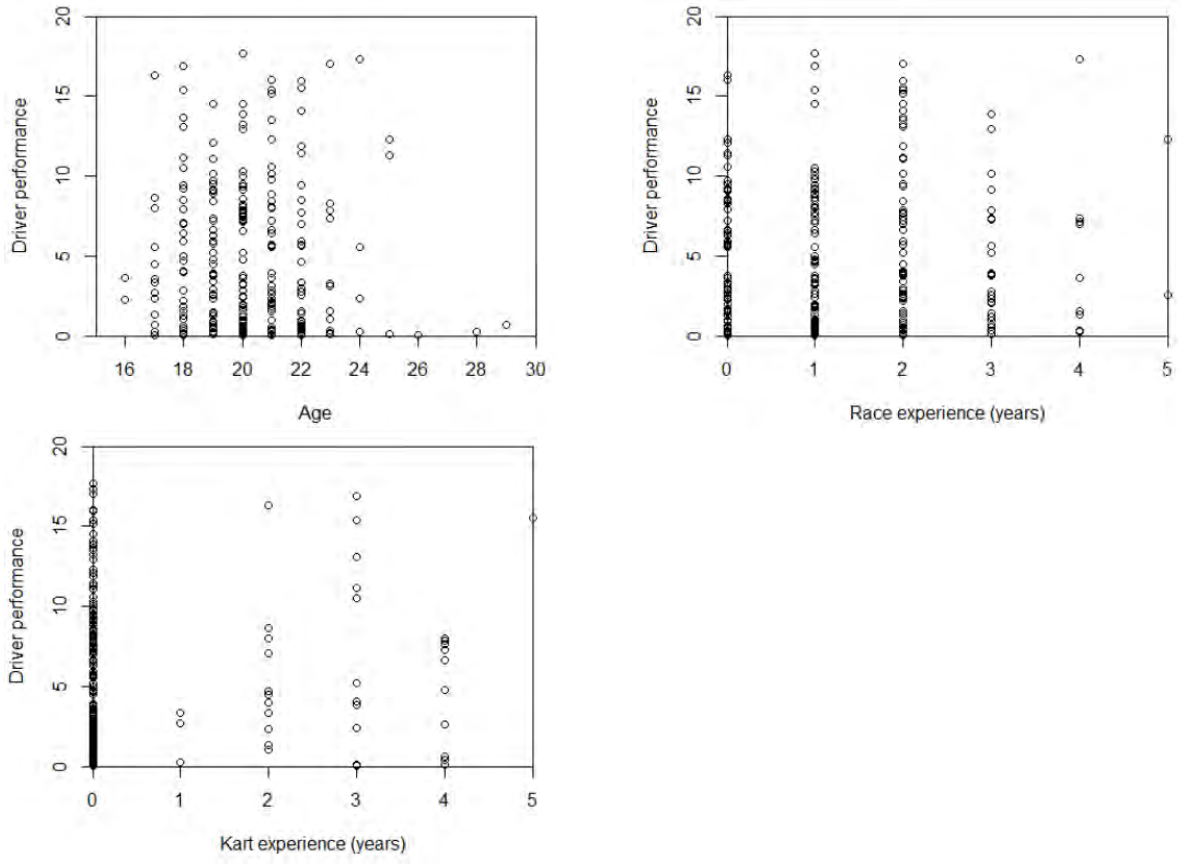


Figure 7: Scatter plots of the features age, race experience and kart experience against the response variable.

6 Model description

This section describes the machine learning models that will be used to predict the driver performance in the FIA F3 European Championship. First, we will give a formulation of the machine learning problem at hand. Second, we will discuss how the hyperparameters of the machine learnings models are tuned. Then, we will provide some theoretical background of each of the machine learning models as well as the implementation in R. Finally, we will discuss how the machine learners models will be evaluated.

6.1 The machine learning task

The goal of this research is to predict the driver performance in the FIA F3 European Championship based on a driver's career path. The features that are used in this machine learning problem describe a driver's performance in other relevant single-seater auto racing championships and karting. As stated in Section 4, we can distinguish three types of features: general performance features, driver performance features and driver features. The response variable describes the driver performance in the FIA F3 European Championship, quantified as the adjusted points rate. This is summarized in Table 14.

Response variable	Y	A driver's adjusted points rate during a season in the FIA F3 European Championship.
Features	$X_1, \ldots, X_k$	General performance features Driver performance features Driver features

Table 14: Formulation of the machine learning problem at hand.

The machine learning problem can be formulated as a regression problem, since the response variable Y is continuous. As a result, we consider machine learning models that are suitable to perform regression tasks. The machine learning models should also be able to handle missing values because of the sparseness of our data set. As already stated in Section 5, one of the challenges of the data set at hand is the large amount of missing values. One method to solve the problem of missing values is to impute these missing values with some imputation method. However, this is not desirable here because then we have to impute the majority of the data points. We thus select machine learning models that have an intrinsic method available to handle the missing values. Another challenge is the large number of features relative to the number of observations. As a result, selecting the features that are expected to have to largest predictive power can be a complex process. We would like to use machine learning algorithms that facilitate the feature selection process. Regression trees, random forests, and gradient boosting models can be used for regression problems, are able to handle missing values and facilitate in the feature selection process. We will discuss some theoretical background of these models in sections 6.3, 6.4, and 6.5. Section 6.6 discusses how the performance of the machine learning models is evaluated. The best performing model will be used to predict the performance of potential F3 drivers in the FIA F3 European Championship. Based on these predictions, we can create a ranking of potential F3 drivers and recommend a talented driver for next season.

6.2 Hyperparameter tuning

All three machine learnings models rely on a certain set of hyperparameters. These hyperparameters cannot be learned by the machine learning models but should be specified in advance. The values of the hyperparameters can affect the performance of the machine learning algorithm significantly, so finding the 'right' values for the hyperparameters is important. This process is called (hyper)parameter tuning. We use the R package *mlrHyperopt* (Richter, 2017) to tune the hyperparameters of the machine learning

models. *MlrHyperopt* tries to simplify hyperparameter tuning of machine learning models by providing suitable search spaces. The following is available for the most common machine learning methods:

- the set of hyperparameters that should be tuned,
- the range within these hyperparameters should be tuned (search space),
- the tuning method that should be used,
- the performance measure that should be used during the tuning process, and
- the resampling method that should be used during the tuning process.

The *mlrHyperopt package* uses the machine learning methods and tuning methods as implemented in the *mlr* package. The *mlr* package (Bischl et al., 2016) provides an interface to the most common classification and regression techniques used in machine learning. It provides the user with a machine learning infrastructure that facilitates hyperparameter tuning, the feature selection process, pre- and post-processing of data, and comparing the performance of different machine learnings models. However, search spaces are not available in the package *mlr*, which makes parameter tuning using this package rather difficult. This problem is solved by the *mlrHyperopt* package by offering a web service to share, upload and download improved search spaces. Table 15 gives an overview of the control parameters that should be specified in order to use the function *hyperopt* from the *mlrHyperopt* package for hyperparameter tuning. The value of these control parameters can be requested from the *mlrHyperopt* package for the most common machine learning methods.

Control parameters	**Description**
mlr.control	Control object for search method. This object selects the optimization algorithm for tuning. The tuning methods available are grid search, random search and Bayesian optimization.
resampling	The resampling method determines how the performance is obtained during tuning. The resampling methods available are 5 or 10 fold cross-validation, leave-one-out cross validation, repeated cross-validation, out-of-bag boostrap, subsampling, holdout data set, growing window cross-validation, and fixed window cross validation.
measures	Performance measure(s) to evaluate. Default is the default measure for the task.
par.config	Defines a parameter configuration that defines the search range for the hyperparameter optimization.

Table 15: Hyperparameter tuning control.

We use the same resampling method, namely 5 fold cross validation, for all machine learning models. The function *hyperopt* uses a heuristic that decides the tuning method.

- Grid search: 1 parameter, 2 mixed parameters.
- Random search: more than 2 mixed parameters.
- Bayesian Optimization with *mlrMBO*: all parameters numeric.

Since we have multiple parameters that are all numeric, Bayesian optimization with *mlrMBO* is used. The function *hyperopt* uses the mean-square error as performance measure, which is the default for regression tasks. The optimal hyperparameters for each machine model as determined by the function *hyperopt* will be provided in Section 7.

6.3 Regression trees

Decision trees are commonly used supervised learning algorithms that use a tree structure to predict the response variable of observations in a data set. We can distinguish between two types of decision trees: classification trees and regression trees. Classification trees can be used to predict a discrete response variable. The nodes of the tree represent the class labels and the branches represent conjunctions of features that result in those class labels. Predictions for future observations are made by using the most common class label among the observations in a terminal node. Regression trees also consists of nodes and branches, but they can be used to predict a continuous response variable. They use the average value of the response variable of observations in a terminal node to predict future observations. Classification And Regression Tree (CART) analysis is a term used to refer to both of the above procedures, first introduced by Breiman, Friedman, Stone, and Olshen (1984).

6.3.1 Theoretical background

We will now focus on how to build a regression tree using the CART method. The regression tree is grown using recursively binary splitting, which means that the data set is recursively split into two subgroups based on a certain splitting criterion. The splitting criterion measures the error on the training data and decides which variable minimizes this error and, thus, gives the best split. The regression tree uses the sum of squared errors $\sum_i (y_i - \bar{y})^2$ of the observations in a node as splitting criterion. The splitting process is repeated until either the number of observations in the subgroups is less than a certain minimum or the result cannot be further improved. The regression tree then averages the value of the response variable of the observations in a terminal node in order to make predictions. Figure 8 gives an example of how such a regression tree would look like when applied to our data set. As can be seen in this figure, the tree is build by splitting the data set repeatedly in two subgroups. First, the tree splits on the performance in the FIA F3 European Championship (first season) by looking at the average number of championship points that a driver has scored. Observations of drivers that scored ten or less points on average are send to the left branch of the tree, while observations of drivers that score more than ten points are send to the right. Then the left branch splits on the performance in the Eurocup Formula Renault 2.0 championship, measured by the number of wins, while the right branch splits on the performance in karts, measured by the final position in the championship standing, and on the age variable.

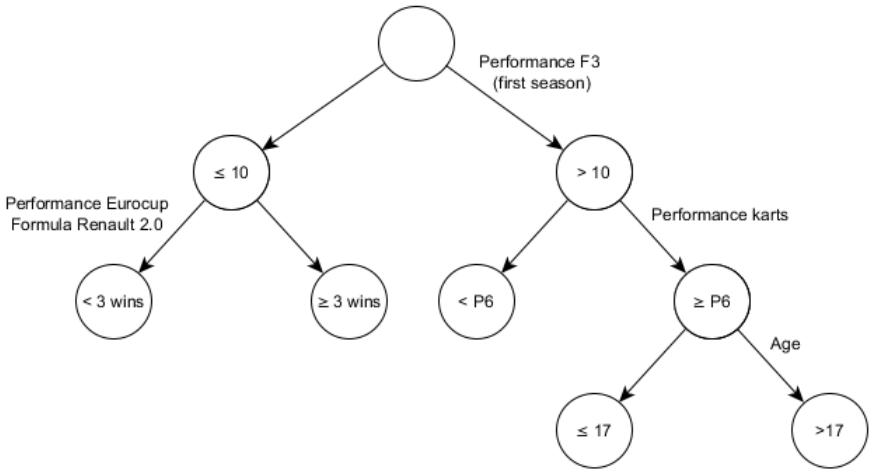


Figure 8: Example of a regression tree.

The variables to split on as well as the split points are determined by the algorithm itself. When the splitting process is completed, the algorithm makes predictions by averaging the value of the response variable of observations in a terminal node.

One important consideration when building a regression tree is how large we should grow the tree; a very large tree might over fit the data, while a small tree might not capture the important structure of the data. According to Friedman, Hastie, and Tibshirani (2001), the preferred strategy is to grow a large tree, stopping the splitting process only when some minimum node size is reached, and then pruning this large tree using cost-complexity pruning, which we will now discuss. This discussion is based on T. M. Therneau, Atkinson, et al. (1997).

Suppose that we have built a complete tree T. This complete tree is likely to over fit the training data, so we use cost-complexity pruning to prune the tree and obtain a model that can be generalized. Define the cost-complexity function of tree T as

$$C_\alpha(T) = R(T) + \alpha|T| \tag{6}$$

where $|T|$ is the number of nodes in tree T and $R(T)$ a loss function calculated across these nodes equal to the sum of squared errors for regression trees. The cost-complexity function incorporates a penalty term into the loss function, to prevent the tree from growing to large and overfitting on the training data. The complexity parameter α controls the trade-off between the size of the tree and the goodness of fit to the training data. Large values of α result in smaller trees T_α, while smaller values of α result in larger trees. As notation suggest, with $\alpha = 0$ the solution is the full tree T_0 and with $\alpha = \infty$ the solution is the tree with no splits. We define a subtree $T \subset T_0$ to be any tree that can be obtained by pruning T_0, that is, collapsing any number of its internal nodes. The idea is to find, for each α, the subtree $T_\alpha \subseteq T_0$ to minimize $C_\alpha(T)$. For each α one can show that there is a unique smallest subtree T_α that minimizes $C_\alpha(T)$. We refer to Breiman et al. (1984) for a discussion on how the optimal value of α and corresponding subtree T_α can be determined.
The R package *rpart* that we will use to build the regression tree uses cross-validation to determine the optimal value of α. A cross validated estimate of the loss function is computed for a nested set of sub trees; the final model is that sub tree with the lowest estimate of the loss function.

6.3.2 Implementation in R

The R package *rpart* (T. Therneau, Atkinson, & Ripley, 2015) uses recursive binary splitting to build classification, regression and survival trees. It contains an implementation of most of the functionality of the 1984 book by Breiman et al. The trees are built using a two-stage procedure. In the first stage recursive binary splitting is used to build up the tree, while in the second stage the tree is pruned using cost-complexity pruning, as discussed in the previous section.

The *rpart* package uses the ANOVA method for building the regression tree with corresponding splitting criteria

$$\text{SST} - (\text{SSL} + \text{SSR}),$$

where $\text{SST} = \sum_i (y_i - \bar{y})^2$ is the sum of squared errors of observations in a node, and SSR, SSL are the sum of squared errors for the right and left son, respectively. The variable and split point that maximizes the value $SST - (SSL + SSR)$ are chosen in each node.

The *rpart* package contains its own method to handle missing values in the data set by constructing so-called surrogate variables. This method allows the algorithm to include any observation in the model with values for at least one feature. To build up the tree, the algorithm computes the splitting criterion using only the observations for which a feature is not missing. Then the algorithm chooses the best primary feature and split point and construct a list of surrogate features and split points. The first surrogate is the feature and corresponding split point that best imitates the split of the training data achieved by the primary split. The second surrogate is the feature and corresponding split points that does this second best, etc. When the primary feature is missing during training or predicting unseen data, the surrogate

splits are used in order. Surrogate splits use the correlation between features to reduce the effect of missing values in the data set. The higher the correlation between the feature that is missing and the other features, the smaller the loss of information due to the missing value.

6.3.3 Hyperparameters

The *rpart* function allows us to specify several hyperparameters. Table 16 contains an overview of the hyperparameters that can be optimized as well as the range that is used while tuning these parameters.

	Description	**Range**	**Default**	**Tuned**
min.split	The minimum number of observations that must exist in a node in order for a split to be attempted.	5 to 50	20	yes
min.bucket	The minimum number of observations in any terminal node.	5 to 50	7	yes
cp	Complexity parameter.	-10 to 0	-6.64	yes
maxsurrogate	The maximum number of surrogate variables to retain at each node.	-	5	no
usesurrogate	How to use surrogates in the splitting process.	-	2	no
xval	Number of cross-validations.	-	10	no
surrogatestyle	Controls the selection of the best surrogate.	-	0	no
maxdepth	Set the maximum depth of any node of the final tree.	3 to 30	30	yes

Table 16: Parameters used by *rpart*.

The parameter *cp* can be used to control the value of the complexity parameter α. This parameter is based on a scaled version of (6):

$$R_{\text{cp}}(T) = R(T) + cp\,|T|\,R(T_1)$$

where T_1 is the tree with no splits, $|T|$ is the number of variables (splits) in a tree, and R represents the loss function. For regression trees, the scaled *cp* can be interpreted as follows: if any split does not increase the overall fit of the model by at least *cp* then that split is considered to be, a priori, not worth pursuing. The algorithm does not split that branch further, which reduces the computational effort considerably. The optimal value of *cp* is determined using cross-validation.

The parameters *maxsurrogate*, *usesurrogate* and *surrogatestyle* can be used to control how the algorithm handles missing values. The parameter *usesurrogate* controls how to use surrogate variables in the splitting process. We set this parameter equal to 2 which means that observations are sent in the majority direction if all surrogates are missing. This is recommended by Breiman et al. (1984). The other options are that the surrogate variables are not used at all (0) or that the observations are not sent further downwards if all surrogates are missing (1). Finally, the parameter *surrogatestyle* controls the selection of the best surrogate. If set to 0 (default) the algorithm uses the total number of correct classifications for a potential surrogate variable, if set to 1 it uses the percentage correct, calculated over the non-missing values of the surrogate. We set *surrogatestyle* to 0, which more severely penalizes covariates with a large number of missing values.

6.3.4 Problems of trees

We now discuss some problem of trees that are mentioned in Friedman et al. (2001). The most important problem of trees is their high variance: a small change in the data can result in very different splits. This instability is caused by the hierarchical nature of the splitting process: an error in the top of the tree is propagated down to all splits below. Another disadvantage is the lack of smoothness in the predictions. This can result in many drivers having the same predicted performance which can make it difficult to

produce sensible rankings. Finally, trees are known to have relatively low predictive power and generate inaccurate results when they are used for predictive learning. Boosting and bagging techniques are known to improve the predictive power by combining the predictions of a large number of regression trees. These techniques are discussed in Section 6.4 and Section 6.5, respectively.

6.4 Random Forest

Random forests are ensemble learning methods for classification and regression tasks. Ensemble learning methods combine multiple machine learnings models to obtain better predictive performance than could be obtained with only one of the models. Two well-known ensemble methods are boosting and bootstrap aggregating, also known as bagging. Boosting methods incrementally build an ensemble by training each new model on the residuals of the previous models (regression) or on the misclassified training examples (classification). Boosting is described in more detail in Section 6.5. Bagging methods weight the predictions of multiple models equally in order to improve the prediction. Bagging tries to reduce the variance of an estimated prediction function and seems to work especially well for high-variance, low-bias procedures, such as decision trees. We will discuss bagging techniques, and, in particular, random forests in more detail in the next section.

6.4.1 Theoretical background

Random forests are ensemble methods that use bagging techniques by combining a large number of random decision trees. These decision trees are fitted on various subsamples of the training data that are drawn with replacement in a bootstrap setting. For regression problems, the predictions of the different decision trees are averaged to obtain a final prediction. This is illustrated in Figure 9.

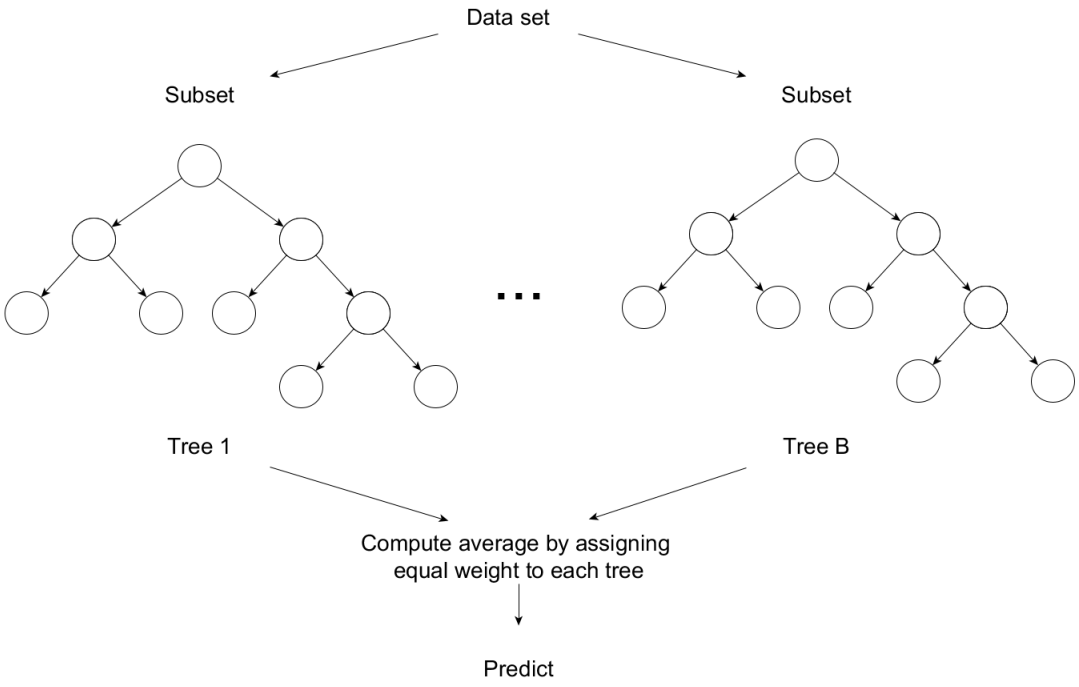


Figure 9: The random forest model illustrated.

Random forests use two important properties of decision trees, namely that they suffer from high variance and, on the other hand, that they have low bias. Moreover, the trees in a random forest are identically distributed, which means that the the expectation of an average of B trees is the same as the expectation of only one tree. Thus, the bias of bagged trees is the same as that of the individual trees, but the variance is reduced through random selection of the input variables. Specifically, when growing a tree on a bootstrapped data set $m \leq p$ of the features are randomly selected as candidates for splitting. Reducing the number of features randomly sampled at candidates at each split m will reduce the correlation between any pair of trees in the ensemble, and hence reduce the variance of the ensemble model.

After B trees are grown on a bootstrapped data set, the random forest predictor for regression is given by

$$\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x; \Theta_b),$$

where $T(x; \Theta_b)$ represents a single tree in the random forest with parameter set Θ_b. This parameter set consists of the features that the tree has chosen to use in the splitting process, the split points, and the predictions of the terminal nodes.

6.4.2 Implementation in R

The package *randomForestSRC* (Ishwaran & Kogalur, 2017) implements Breiman's random forest algorithm (Breiman, 2001) for survival, classification and regression forests. The function *rfsrc* can be used to grow a random forest using training data. The *na.impute* option of this function *rfsrc* imputes missing data, both in the features and the response variable, using a modification of the missing data algorithm of Ishwaran, Kogalur, Blackstone, Lauer, et al. (2008). The algorithms imputes missing values of a feature by randomly drawn values from non-missing data of that feature before splitting. Because of the imputed data, it is possible to assign data points to left and right nodes when the node is split on a feature with missing data. However, the splitting criterion is only computed using non-missing data, not imputed data. After a node split, the imputed data is replaced again by missing data and the process is repeated until the terminal nodes are reached. Missing values in terminal nodes are imputed using non-missing data from the terminal node. Categorical features are imputed by the mode, while numeric features are imputed by the mean. Finally, the algorithm removes observations that have only missing features and a missing response variable. Variables having all missing values are also removed.

6.4.3 Hyperparameters

The *rfsrc* function allows us to specify several hyperparameters. Table 17 contains an overview of the hyperparmeters than can be optimized as well as the range that is used while tuning these parameters.

	Description	**Range**	**Default**	**Tuned**
ntree	Number of trees in the forest.	$100, 200, 300, 400, 500$	-	yes
mtry	Number of variables randomly selected as candidates for splitting a node.	1 to p	$p/3$	yes
nodesize	Forest average number of unique cases (data points) in a terminal node.	1 to 10	-	yes
nodedepth	Maximum depth to which a tree should be grown.	1 to 10	-	yes

Table 17: Parameters used by *rfsrc*, where p is equal to the number of features.

In general, the value of *ntree*, the number of trees in the forest, should not be set too small to ensure that every observation is predicted at least a few times. On the other hand, the value of *mtry* should not set too large because a small m increases the variance reduction. We have chosen the parameter configuration ourselves, because *mrlHyperopt* has no search spaces available for the *randomforestSRC* package. If search spaces for the *randomforestSRC* will be available in the future, the parameter tuning may be improved.

6.5 Gradient boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak learners, typically decision trees. A weak learner is a learning algorithm whose predictive performance is only slightly better than random guessing. As stated before, boosting methods for regression incrementally build an ensemble by training each new model on the residuals of the previous models. This is illustrated in Figure 10. Next, predictions are made by computing a weighted average of the regression trees, where observations that are more difficult to predict are assigned more weight.
We will now discuss Friedman's gradient boosting machine as the R package that we will use, *gbm*, closely follows this implementation.

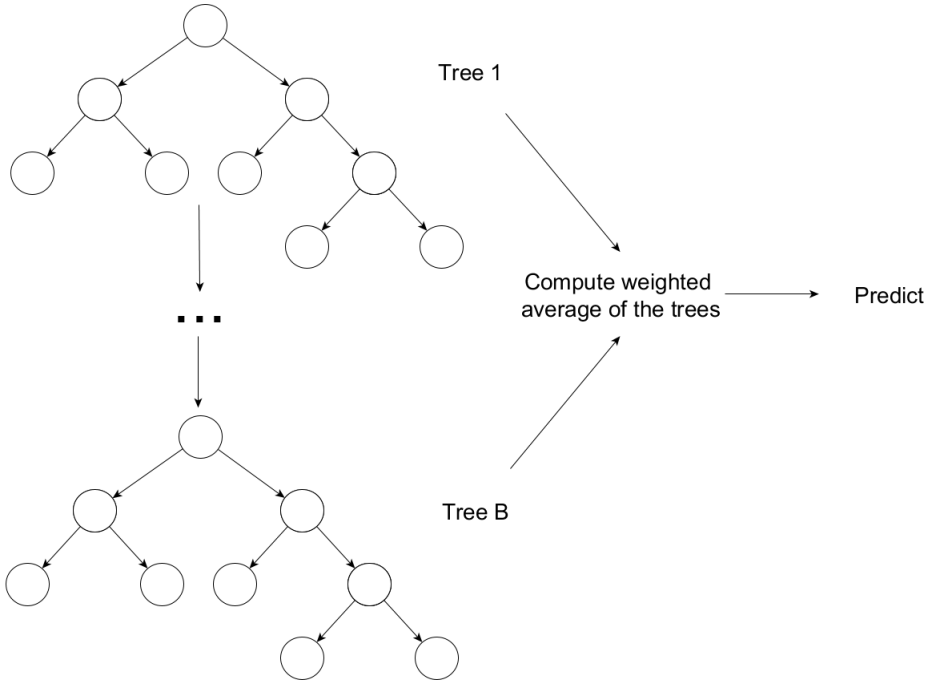


Figure 10: The gradient boosting model illustrated.

6.5.1 Theoretical background

This section discusses some theoretical background of Friedman's gradient boosting machine. We closely follow the book of Friedman et al. (2001).

Consider a regression problem where we want to find a regression function $f(x)$ that minimizes some loss function $L(f)$ on the training data

$$L(f) = \sum_{i=1}^{N} L(y_i, f(x_i)), \tag{7}$$

where x_i is the feature vector of observation i, y_i is the value of the response variable of observation i, and N is the number of observations. In a regression context, the loss function is generally given by the squared error or the absolute error.

Boosting techniques fit an additive expansion in a set of elementary 'basis' functions to the training data. Basis function expansions take the form

$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m) \tag{8}$$

where β_m, $m = 1, \ldots, M$ are the expansion coefficient, and $b(x; \gamma) \in \mathcal{R}$ are usually simple functions characterized by a set of parameters γ. These simple functions are the individual prediction models, for example the decision trees. For decision trees, γ parametrizes the features to split on and split points at the internal nodes, and the predictions at the terminal nodes. These individual models are fit by minimizing the loss function of (7) on the training data,

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b(x_i; \gamma_m)\right). \tag{9}$$

In case of the boosted tree model, we can write the function $f(x)$ as the sum of M regression trees

$$f_M(x) = \sum_{m=1}^{M} T(x, \Theta_m)$$

where $T(x, \Theta_m)$ represents a regression tree with parameter set Θ_m consisting of the splitting features and points. The idea of gradient boosting is to find the tree that maximally reduces

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m)),$$

given the current model f_{m-1} and its fit $f_{m-1}(x_i)$. The tree $T(x_i; \Theta_m)$ will be trained to minimize the difference between the target function $f(x)$ and the current prediction of the model by reconstructing the residual. At each iteration the algorithm determines the direction, the gradient, in which it needs to improve the fit to the data. Then a regression tree is fit to the components of the negative gradient, which are referred to as generalized or pseudo residuals. Finally, the boosted model found so far is updated.

6.5.2 Implementation in R

The R package *gbm* (Ridgeway, 2017) implements Friedman's Gradient Boosting Machine (Friedman, 2001), as discussed in the previous section. This package includes a variety of regression methods such as regression methods for least squares, absolute loss, t-distribution loss, quantile regression, logistic, multinomial logistic, and Poisson. We choose the distribution equal to 'Gaussian'' in order to use least squares regression.

6.5.3 Hyperparameters

The *gbm* function allows us to specify certain hyperparameters. Table 18 contains an overview of the hyperparameters than can be optimized as well as the range that is used while tuning these parameters. An important consideration when using the *gbm* package is which values the parameters *n.trees* and *shrinkage* should have. The parameter *n.trees* represents the number of boosting iterations or the number of trees to fit. Each iteration usually reduces the loss function on the training set, such that for *n.trees* large enough this loss will be arbitrarily small. However, this can lead to overfitting which has a negative influence on the out-of-sample predictive performance of the model. The *shrinkage* parameter is used to scale the contribution of each tree by a certain factor when it is added to the current model. This parameter can be used to control the learning rate of the boosting procedure. There exists a trade-off between the parameters *n.trees* and *shrinkage*: smaller shrinkage values lead to larger *n.trees* values for the same loss value on the training data. Empirically it has been shown that smaller shrinkage values result in smaller test set errors. However, smaller shrinkage values will also result in higher computational costs in both storage and CPU time.

	Description	**Range**	**Default**	**Tuned**
n.trees[1]	The total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion.	0 to 6.64	5.64	yes
interaction depth	The maximum depth of variable interactions.	1 to 10	1	yes
n.minobsinnode	Minimum number of observations in the trees terminal nodes.	5 to 25	10	yes
shrinkage	A shrinkage parameter applied to each tree in the expansion, also known as the learning rate or step size reduction.	0.001 to 0.6	0.001	yes
bag.fraction	The fraction of the training set observations randomly selected to propose the next tree in the expansion.	-	0.5	no
cv.folds	Number of cross-validation folds to perform.	-	0	no

Table 18: Parameters used by *gbm*.

6.6 Model evaluation

We randomly split the data set into two parts: a training set (80%) and a test set (20%). The training set is used to train the machine learning models and to tune the hyperparameters. We use the R package *mlrHyperopt* to facilitate the hyperparameter tuning, as discussed in Section 6.2. The test set is used to evaluate the performance of the machine learnings models. We use three evaluation metrics for this purpose: the root-mean-square error (RMSE), Spearman's correlation coefficient, and the normalized discounted cumulative gain (nDCG). These evaluation metrics are discussed in Section 6.6.1, 6.6.2, and 6.6.3.

However, testing the performance on only one test set can result in an estimate of the performance that is too optimistic or too pessimistic depending on the structure of that specific test set. We thus prefer to evaluate the performance of the models on different test sets rather than only one test set. This is achieved by splitting the data set repeatedly, say B times, in a training- and a test set using a bootstrap procedure. Each time, the model is trained on the training set using the optimal hyperparameters, and the performance is evaluated on the test set. This produces a sample of B values for each performance measure

[1]We assume the range and the default value of the *n.trees* parameter to be wrongly documented by the *mlrHyperopt* package. The number of trees should be considerably large and the optimal values that were obtained by the tuning method were out of the reported range.

that can be used to construct bootstrap confidence intervals. Figure 11 summarizes the evaluation method that is used in this research.

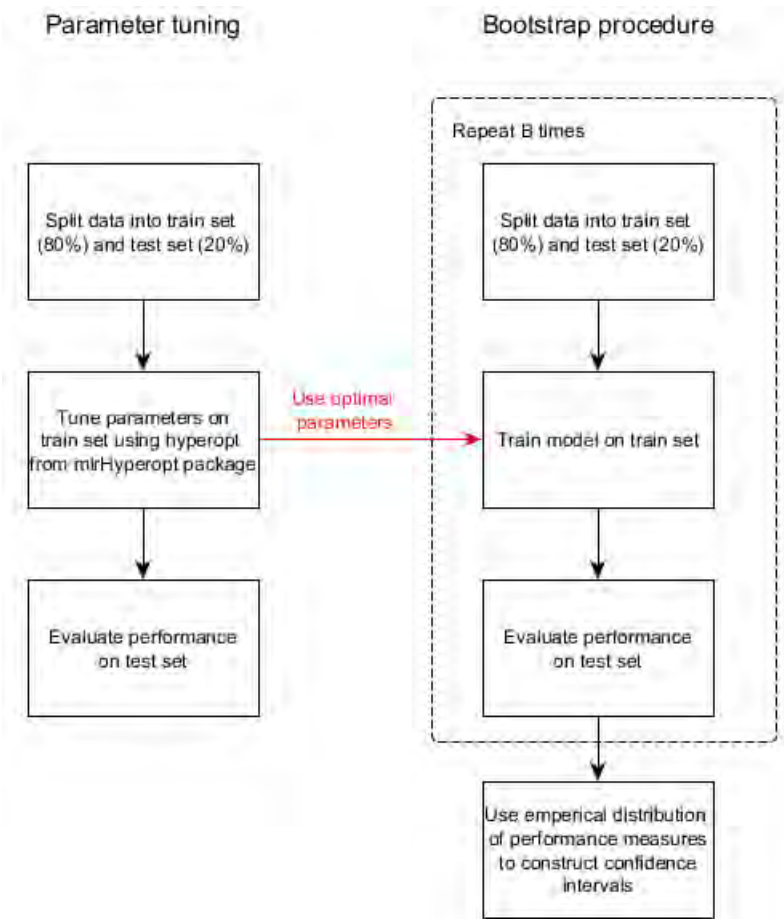


Figure 11: Evaluation method: the optimal hyperparameters as found by the *mlrHyperopt* package are used to evaluate the machine learnings models in a bootstrap procedure.

6.6.1 Root-mean-square error

We evaluate the performance of the machine learning models by computing the root-mean-square error (RMSE), which is a statistic that measures the difference between the predictions and the actual values of the response variable. The root-mean-square error can be computed as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)}{N}},$$

where $\hat{y}_i$ is the prediction of the response variable of observation i, y_i is the actual value of the response variable of observation i, and N is the total number of observations.

6.6.2 Spearman's correlation coefficient

Spearman's correlation coefficient computes the rank correlation between the predictions and the actual values of the response variable. Let rg_{y_i} and $rg_{\hat{y}_i}$ be the ranks of the values of the response variable and the predictions, respectively. Then Spearman's rank correlation coefficient is computed as the Pearson correlation coefficient between the rank variables:

$$r_s = \rho_{rg_{y_i} rg_{\hat{y}_i}} = \frac{\text{cov}(rg_{y_i}, rg_{\hat{y}_i})}{\sigma_{rg_{y_i}} \sigma_{rg_{\hat{y}_i}}} \in [-1, 1].$$

6.6.3 Normalized discounted cumulative gain (nDCG)

The normalized discounted cumulative gain (nDCG) is a commonly used evaluation metric in the context of information retrieval. It measures the effectiveness of web search engine algorithms or related applications. Web search engine algorithms are given a set of queries and are supposed to produce a ranking of documents per query in order of relevance. In this context, the nDCG metric is computed by normalizing the discounted cumulative gain (DCG) by the ideal discounted cumulative gain (IDCG) and averaging this value over all queries. Thus, nDCG_k can be computed by

$$nDCG_k = \frac{DCG_k}{IDCG_k},$$

where k is the maximum number of entities that can be recommended.
The discounted cumulative gain (DCG) is a measure of ranking quality. This metric uses a graded relevance scale, which indicates how relevant a document is to the query. The DCG then measures the usefulness or gain of a document based on its position in the result list. The gain is then accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks. Summarizing, the value of DCG_k is computed as

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)},$$

where rel_i represents the relevance grade assigned to the document on position i.
The IDCG is the maximum possible (ideal) DCG for a given set of queries, documents, and relevance grades.

We can use the nDCG to measure the quality of the ranking that is produced by the machine learning model. Note that the ranking of F3 drivers can be considered as one query, so averaging the nDCG over all queries is not needed. First, we need to assign relevance grades to all observations. These relevance grades are based on expert rules that are created in consultation with Van Amersfoort Racing. These rules distinguish between 'rookie' drivers and 'no rookie' drivers. A driver is a 'rookie' driver when he/she competes for the first season in the FIA F3 European Championship. A win is for example more impressive in a driver's rookie year than in this second or third year. The expert rules are based on the results that a driver achieves in a certain season, as can be seen in Table Table 19. Note that a driver can be assigned multiple relevance grades because a driver can correspond to multiple observations in the data set if he competes for multiple season in the FIA F3 European Championship.

Rookie	**No rookie**	**Rel**$_i$
No more than 10% in highest 30% positions.	No more than 20% in highest 30% positions.	1
More than 10% in highest 30% positions.	More than 10% in highest 30% positions.	2
More than 10% podium places.	More than one win.	3
One or more wins.	More than three wins.	4

Table 19: Relevance grades according to the expert rules.

After we assigned the relevance grades to the drivers based on these expert rules, an expert of Van Amersfoort Racing has adjusted the grades that do not agree with his opinion. The resulting list of relevance grades is used to compute the nDCG.

6.6.4 Bootstrap procedure

The performance of the machine learning models is evaluated using a bootstrap procedure. We split the data 100 times in a training set (80%) and a test set (20%). Each time the machine learning models are trained and the performance is evaluated on the test set. This procedure gives us a sample containing 100 values for each performance measure and for each machine learning model. The value of the performance measures is then estimated by the sample mean.

Denote the true value of the performance measure by θ and the sample mean by T. We can use bootstrap methods to express the accuracy of T. Consider the following bootstrap scheme

1. Simulate $B = 1000$ independent bootstrap samples $X_1^*, ..., X_n^*$ from the sample of the performance measure.
2. Compute for each of the B bootstrap samples the sample mean, $T_n^* = \frac{1}{n}\sum_{i=1}^{n} X_i^*$; call the resulting values $T_{n,1}^*, ..., T_{n,B}^*$.

The empirical distribution $T_{n,1}^*, ..., T_{n,B}^*$ can be used to express the accuracy of T in two ways:

1. Use the empirical distribution $T_{n,1}^*, ..., T_{n,B}^*$ to estimate the variance of T by computing the sample variance.
2. Use the empirical distribution $T_{n,1}^*, ..., T_{n,B}^*$ to compute a bootstrap confidence interval. A basic bootstrap interval can be computed using the formula

$$2T - \hat{\theta}*_{([B(1-\alpha)])}, 2T - \hat{\theta}*_{[B\alpha]}$$

We use the R package *boot* to perform the bootstrap sampling and to calculate bootstrap confidence intervals.

7 Results

This section presents the results of this research. First, we provide the optimal hyperparameters of each machine learning model as found using the package *mlrHyperopt*. Then, we evaluate the performance of each model on a held-out test set. We also evaluate the performance of the models on different test sets in a bootstrap fashion and express the accuracy of the estimators using a bootstrap estimator of the standard deviation and by constructing bootstrap confidence intervals.

7.1 Hyperparameters

Table 20 contains the optimal hyperparameters of each machine learning model as determined by the function *hyperopt* of the *mlrHyperopt* package. We also provide the mean RMSE for each machine learning model obtained during cross-validation. The optimal hyperparameters will be used to obtain the results in Section 7.2.

Regression tree		**Random forest**		**Gradient boosting**	
cp	0.0019	ntree	100	ntrees	202
maxdepth	8	mtry	192	shrinkage	0.0429
minbucket	8	nodesize	1	interaction.depth	1
minsplit	23	nodedepth	10	n.minobsinnode	9
		nsplit	72		
		nimpute	1		
mean RMSE	3.832	*mean RMSE*	3.710	*mean RMSE*	3.469

Table 20: Optimal hyperparameters determined using the *MLHyperOpt* package.

7.2 Model results

We will now evaluate the performance of the machine learning models using the performance measures described in Section 6.6.1, 6.6.2, and 6.6.3.

Table 21 shows the performance of the machine learning models on a held-out test set. These results show that the random forest model performs best on the held-out test set in terms of the RMSE and Spearman's correlation coefficient. However, the value of the nDCG is similar to that of the gradient boosting model. Moreover, the gradient boosting model does not really outperform the regression tree on this particular test set. This result is not expected, as theoretically the gradient boosting model should outperform a single regression tree.

Model	**RMSE**	**Spearman correlation**	**nDCG**
Regression tree	4.311	0.506	0.859
Random forest	3.034	0.620	0.883
Gradient boosting	4.267	0.508	0.885

Table 21: Performance of the machine learning models on a held-out test set (randomly drawn).

The results of the bootstrap procedure are more informative because they reflect the performance of the machine learning models on different test sets. Tables 22, 23, and 24 present the mean values of the RMSE, Spearman's correlation coefficient, and the nDCG, respectively, for all three machine learnings models. These tables also contains a bootstrap estimator of the standard deviation and a 95% bootstrap confidence interval. These can be used to quantify the accuracy of the sample mean as an estimator for the mean performance.

Based on these results, we can conclude that the gradient boosting model performs best on average in terms of the RMSE and Spearman's correlation coefficient. However, we obtained a higher mean nDCG using the random forest model.

The estimated bootstrap standard error and the 95% bootstrap confidence interval can be used to quantify the accuracy of the sample mean as estimator of the underlying performance. These results show that the estimated bootstrap standard errors are similar for all three machine learning models. We can observe some differences between the performance measures implying that there is higher variation in the estimator of the RMSE than in the estimator of Spearman's correlation coefficient and the nDCG. This is expected since Spearman's correlation coefficient and the nDCG are bounded, while the RMSE is not. It is difficult to draw any other conclusions on the accuracy of the sample mean as estimator for the underlying performance because we cannot compare the estimates of the standard error and the 95% bootstrap confidence interval to related research. We included these results mainly for the sake of completeness and for comparison with future research.

	Mean	**Bootstrap standard error**	**95% bootstrap confidence interval**
Regression tree	3.977	0.039	$(3.903, 4.057)$
Random forest	3.606	0.037	$(3.531, 3.676)$
Gradient boosting	3.530	0.037	$(3.452, 3.600)$

Table 22: Performance of the machine learning models estimated on the bootstrap sample as the sample mean of the RMSE.

	Mean	**Bootstrap standard error**	**95% bootstrap confidence interval**
Regression tree	0.496	0.010	$(0.4751, 0.5171)$
Random forest	0.574	0.009	$(0.5567, 0.5913)$
Gradient boosting	0.592	0.009	$(0.5756, 0.6100)$

Table 23: Performance of the machine learning models estimated on the bootstrap sample as the sample mean of Spearman's correlation coefficient.

	Mean	**Bootstrap standard error**	**95% bootstrap confidence interval**
Regression tree	0.837	0.005	$(0.8285, 0.8466)$
Random forest	0.864	0.006	$(0.8528, 0.8750)$
Gradient boosting	0.849	0.005	$(0.8389, 0.8588)$

Table 24: Performance of the machine learning models estimated on the bootstrap sample as the sample mean of the nDCG.

We now perform statistical tests to investigate whether there exists a significant difference in the performance of the models. The Shapiro-Wilk test is used to test the samples of the performance measures for normality. This test does not reject the null hypothesis of normality for the RMSE samples, from which we can conclude that there is no reason to assume that those samples are not normally distributed. However, the null hypothesis is rejected for the samples containing Spearman's correlation coefficient and the nDCG. Based on these results, we use a t-test to test for a significant difference in the mean RMSE and a Wilcoxon signed rank test to test for a significant difference in the mean of the other performance measures. Both tests are performed for paired-samples since the performance of each model is evaluated on the same test sets. The results can be found in Table 25 and show that the random forest model and

gradient boosting model outperform the regression tree significantly based on the RMSE and Spearman's correlation coefficient for any reasonable significance level. However, the conclusions of the other tests depend on the significance level that is used. We can conclude that both the random forest model and the gradient boosting model outperform the regression tree based on the nDCG using a significance level of 0.1. However, using a significance level 0.05 would not have resulted in the same conclusion. This also holds if we test for a significant difference in the performance of the random forest model and the gradient boosting model. We find a statistical difference in the performance of the random forest model and gradient boosting model based on the RMSE and Spearman's correlation coefficient using a significance level of 0.1. Based on this result, we can conclude that the gradient boosting model significantly outperforms the random forest model based on the RMSE and Spearman's correlation coefficient. Using the lower significance level of 0.05, however, would not have resulted in the same conclusion. Based on the nDCG, we can conclude that the random forest model performs better than the gradient boosting model because the null hypothesis is rejected for any reasonable significance level.

Performance measure	**Machine learning models**		**Alternative**	**p-value**
RMSE	Random forest	Regression tree	less	1.060×10^{-9}*
	Gradient boosting	Regression tree	less	6.450×10^{-14}*
	Gradient boosting	Random forest	less	0.060
Spearman's cor. coef.	Random forest	Regression tree	greater	1.311×10^{-8}*
	Gradient boosting	Regression tree	greater	3.197×10^{-10}*
	Gradient boosting	Random forest	greater	0.080
nDCG	Random forest	Regression tree	greater	6.009×10^{-5}*
	Gradient boosting	Regression tree	greater	0.062
	Random forest	Gradient boosting	greater	0.005*

Table 25: Results of the t-test and the Wilcoxon signed rank test. Testing for a difference in performance of the machine learning models. * denotes rejection at 0.05 significance level.

We choose the gradient boosting model as our final model based on the results presented in this section. The gradient boosting model does significantly outperform the random forest model on the RMSE and Spearman's correlation coefficient using a reasonable significance level of 0.1. However, we obtained a significantly higher nDCG using the random forest model. These conclusions seem contradictory, but we prefer to measure the performance of the models using the RMSE and the Spearman's correlation coefficient rather than using the nDCG. We use the nDCG to measure the quality of only one query (the ranking of F3 drivers) instead of multiple queries, and, as a result, we believe that the nDCG is not able to reflect the difference in performances of the models. Finally, we have to mention that the gradient boosting model is known to usually outperform the random forest model. This is because the gradient boosting model tries to reduce the bias and the variance, while the random forest model only tries to reduce the variance of the predictions. Therefore, the gradient boosting model will generally produce more accurate predictions, but this depends on the data set that is used.

7.3 Final model

We choose the gradient boosting model as our final model, based on the results and the motivation of the previous section. This model consists of 202 regression trees. In total there are 982 predictors of which 52 has non-zero influence. The RMSE computed on the training set is equal to 2.713. We use the final model to investigate the importance of the features and to produce predictions for unseen data.

7.3.1 Feature importance

Confidential

7.3.2 Prediction in the winter of 2017

Confidential

8 Discussion

The goal of this master thesis was to provide an objective method to select talented drivers that can support Van Amersfoort Racing in the driver selection process. We used a typical machine learning approach. First, we collected race results data of the FIA F3 European Championship and other relevant championships in single-seater auto racing and karting. These data were used to create features for the machine learnings models that describe a driver's career path. The machine learnings models were trained to predict the driver performance in the FIA F3 European Championship based on a driver's career path. Finally, we evaluated the performance of the machine learnings models in order to find a answer to the research question:

How accurately can we predict the performance in the FIA F3 European Championship based on a driver's career path?

The performance of the machine learnings models was evaluated using three performance measures: the RMSE, Spearman's correlation coefficient and the nDCG. Moreover, we used a bootstrap procedure to evaluate the performance of the models on different test sets. The results show that the gradient boosting model performs best on average with a mean RMSE of 3.530, a mean Spearman's correlation coefficient of 0.592, and a mean nDCG of 0.849. We obtained a lower nDCG with the random forest model but we considered the RMSE and Spearman's correlation coefficient to be more important for comparing the performance of the models. Statistical tests have shown that both the random forest model and gradient boosting model perform significantly better than the regression tree using any reasonable significance level. This result was expected because bagging techniques are known to reduce the variance, while boosting techniques are know to both reduce the variance and the bias. Both techniques combine a large number of regression trees to obtain a better predictive performance than could be obtained from any of the regression trees alone. We have also find a significant difference in the performance of the random forest and the gradient boosting model using a significance level of 0.1. Based on these results, we choose the gradient boosting model as our final model. The final model was used to investigate the feature importance. This analysis showed that the most important features are *Team*, *Nationality* and the performance during a driver's rookie year in the FIA F3 European Championship. The feature *Team* was expected to have zero importance because the driver performance in the FIA F3 European Championship is obtained by correcting for team performance. The fact that *Team* has non-zero importance can imply that the statistical model has not fully captured the influence of the team performance on the total performance. Also, performance in the Eurocup Formula Renault 2.0, GP3 Series, and British F3 International Series are considered as important predictors for the performance in the FIA F3 European Championship. Finally, we assumed that it was the winter of 2017 and used the gradient boosting model to produce a ranking of the 2018 F3 drivers. This ranking was compared to the current 2018 championship standings, to assess whether the model produces usable rankings. Note that the championship standings are influenced by team performance and competition effects while the predicted driver performance is not. Thus, one should not use this comparison to draw strong conclusions about the performance of the machine learning models but to get some intuition for the kind of predictions that the model produces.

This research provides a new approach for finding talented drivers because, to our knowledge, using a machine learning approach to predict the driver performance in a certain championship is new in this area of sports. Most racing teams contract their drivers based on race results data and expert opinions, rather than on objective measures of driver performance. Some motor sport new websites provide rankings of junior drivers but these rankings are rather subjective and mainly used to speculate about rising F1 stars. The studies that are most closely related to this research provide rankings of F1 drivers based on some measure of driver performance. However, these rankings are based on actual performance in the F1 championship rather than on predicted performance. A driver that has never competed in a F1 race cannot be ranked by these method. Thus, it is not possible to compare our findings with findings of similar studies because these are not available to our knowledge.

We will now discuss challenges of this research and limitations of the methods that are used. One challenge of this research is the data availability. Open-source data sets that contain the race results data of lower championships than F1 are not available to our knowledge. As a result, collecting the data for this research was a time consuming process because we had to use different sources and fix inconsistencies manually. Adding new data to the machine learning model is for these reasons rather difficult. However, new data is generated during each race weekend and it is essential to add these data to the machine learning model in order to produce up-to-date predictions.

We can distinguish between limitations of the driver performance model and the machine learning models. The driver performance model does not perform as well on our data as on F1 data. This is caused by the fact that drivers and teams often compete for a long period in F1, while this is not the case for lower championships. Moreover, Phillips fitted his model on a large data set containing F1 results from 1950-2013, while we use relatively small data sets. This made it in particular difficult to estimate the team coefficient in a reliable way. We had to correct the team coefficients by the number of observations to compute suitable adjusted points rates. Nevertheless, we have concluded, based on a qualitative analysis, that the driver performance model produced decent results because most of the drivers that were ranked high by the driver performance model were (previous) F1 drivers. The results of the driver performance model were then used in the machine learning model as features and as response variable. Any unreliable adjusted points rates features are not expected to strongly influence the results of the model because if that feature is indeed unreliable then it will probably not be used by the machine learning model. Instead, the non-adjusted points rate will be used. On the other hand, unreliable adjusted point rates in the response variable can influence the final ranking of the drivers. Improving the driver performance model can thus result in more reliable rankings. However, based on the qualitative analysis, we have no indications that the model ranks certain teams or drivers consistently too low or too high. We thus believe that the adjusted points rate is a more reliable measure of driver performance than, for example, the non-adjusted points rate.

Limitations of the machine learning approach include the large amount of missing values in the data set and the limited number of observations relative to the number of features. The fact that the data set contains so many missing values makes the machine learning task at hand rather difficult. We are limited to machine learning models that can handle missing values because imputing the majority of the data set is not desirable. But even for models that can handle missing values the predictive power can be increased when more data is available. The missing values are also a disadvantage because they result in unbalanced features. Some features contain considerably more missing values than others. As a result, features that contain relatively many observations might be considered more important by the machine learnings models. Those features will then in particular be used to make predictions, while other features might contain more predictive power if the number of missing values is reduced. An example of a feature that contains no missing values and is considered important by the model is *Nationality*. A disadvantage of this feature is that the factors are unbalanced. For example, the nationalities 'British' and 'German' occur relatively often in the data set, which can give drivers of these nationalities a higher ranking than they would be given only based on performance. It is important to investigate the influence of *Nationality* on the produced rankings, because the rankings should reflect the driver performance rather than the frequency of the nationalities in the data set. The ranking of the 2018 drivers in the winter of 2017 is therefore produced without the *Nationality* feature. Finally, we have a limited number of observations to train the machine learnings model on because we only consider the FIA F3 European Championship from 2008-2017. Including more data is, however, not desirable because race results data from before 2008 might not be representative to predict the driver performance in F3 during the next season. The number of observations will increase when Van Amersfoort Racing decides to compete in 2019 in a new championship that combines the FIA F3 European Championship and the GP3 Series. In that case, we have to retrain the model on data of F3 drivers and GP3 drivers, which will results in approximately twice as many observations.

9 Conclusion

We can conclude that this research provides a new approach for finding talented drivers in single-seater auto racing. The machine learning model can support Van Amersfoort Racing in the driver selection process by providing them with rankings based on driver performance. These rankings can help Van Amersfoort Racing to find a new driver for next season. We used quantitative as well as a qualitative measures to evaluate the performance of the machine learning models and to answer the research question. The gradient boosting model seems to produce usable predictions but its performance should be carefully evaluated in the future when it is used to find new drivers. The model should be used as a supporting rather than a guiding tool and future predictions should be compared with expert opinions.

Suggestions for future research include improving the driver performance model to obtain more reliable rankings. Moreover, reducing the number of missing values in the data set can improve the predictive power of the machine learning models. The number of missing values can be reduced by adjusting the feature engineering process and, for example, by combining features of similar championships or combining features of different seasons in the same championship. Other features that can be included in the machine learning model are data about sponsorships or talent programs that a driver is associated with, qualifying results, lap time data, telemetry data (data from sensors on the car) or data about overtaking actions. These data also describe the performance of driver in a certain championship, but cannot as easy be collected as race results data. Another suggestion, which might be too ambitious for the current level of technology in lower championships of single-seater auto racing, is the use of image recognition to analyze images or video footage of the driving. Finally, applying the methods of this research to other championships than F3 will provide us with comparable results and can help to improve the performance of the model in the future.

References

Aitken, T. (2004). Statistical analysis of top performers in sport with emphasis on the relevance of outliers. *Sports Engineering*, *7*(2), 75–88.

Anderson, A. (2014). Maximum likelihood ranking in racing sports. *Applied Economics*, *46*(15), 1778–1787.

Bell, A., Smith, J., Sabel, C. E., & Jones, K. (2016). Formula for success: Multilevel modelling of formula one driver and constructor performance, 1950–2014. *Journal of Quantitative Analysis in Sports*, *12*(2), 99–112.

Berry, S. M., Reese, C. S., & Larkey, P. D. (1999). Bridging different eras in sports. *Journal of the American Statistical Association*, *94*(447), 661–676.

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., ... Jones, Z. M. (2016). mlr: Machine learning in r. *Journal of Machine Learning Research*, *17*(170), 1-5. Retrieved from `http://jmlr.org/papers/v17/15-066.html`

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *Classification and regression trees*. Taylor & Francis. Retrieved from `https://books.google.nl/books?id=JwQx-WOmSyQC`

Brown, M., & Sokol, J. (2010). An improved lrmc method for ncaa basketball prediction. *Journal of Quantitative Analysis in Sports*, *6*(3).

Davis, C. (2000). *The best of the best: A new look at the great cricketers and their changing times*. ABC Books for the Australian Broadcasting Corporation.

Eichenberger, R., Stadelmann, D., et al. (2009). Who is the best formula 1 driver?: An economic approach to evaluating talent. *Economic Analysis and Policy*, *39*(3), 389.

Elzhov, T. V., Mullen, K. M., Spiess, A.-N., & Bolker, B. (2016). minpack.lm: R interface to the levenberg-marquardt nonlinear least-squares algorithm found in minpack, plus support for bounds [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=minpack.lm` (R package version 1.2-1)

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1) (No. 10). Springer series in statistics New York, NY, USA:.

Glickman, M. E., & Hennessy, J. (2015). A stochastic rank ordered logit model for rating multi-competitor games and sports. *Journal of Quantitative Analysis in Sports*, *11*(3), 131–144.

Ishwaran, H., & Kogalur, U. (2017). Random forests for survival, regression, and classification (rf-src) [Computer software manual]. manual. Retrieved from `https://cran.r-project.org/package=randomForestSRC` (R package version 2.5.1)

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., Lauer, M. S., et al. (2008). Random survival forests. *The annals of applied statistics*, *2*(3), 841–860.

Johnson, S. G. (2014). The nlopt nonlinear-optimization package.

Kuhn, M., with contributions from Jed Wing, Weston, S., Williams, A., Keefer, C., Engelhardt, A., ... Hunt., T. (2017). caret: Classification and regression training [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=caret` (R package version 6.0-78)

Kvam, P., & Sokol, J. S. (2006). A logistic regression/markov chain model for ncaa basketball. *Naval Research Logistics (NrL)*, *53*(8), 788–803.

Kvam, P. H. (2011). Comparing hall of fame baseball players using most valuable player ranks. *Journal of Quantitative Analysis in Sports*, *7*(3).

Mease, D. (2003). A penalized maximum likelihood approach for the ranking of college football teams independent of victory margins. *The American Statistician*, *57*(4), 241–248.

Phillips, A. J. (2014). Uncovering formula one driver performances from 1950 to 2013 by adjusting for team and competition effects. *Journal of Quantitative Analysis in Sports*, *10*(2), 261–278.

Richter, J. (2017). mlrhyperopt: Easy hyperparameteroptimization with mlr and mlrmbo [Computer software manual]. Retrieved from `https://github.com/jakob-r/mlrHyperopt` (R package version 0.1.1)

Ridgeway, G. (2017). gbm: Generalized boosted regression models [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=gbm` (R package version 2.1.3)

Rudis, B. (2018). splashr: Tools to work with the 'splash' 'javascript' rendering and scraping service [Computer software manual]. Retrieved from `http://github.com/hrbrmstr/splashr` (R package version 0.5.0)

Therneau, T., Atkinson, B., & Ripley, B. (2015). rpart: Recursive partitioning and regression trees [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=rpart` (R package version 4.1-10)

Therneau, T. M., Atkinson, E. J., et al. (1997). *An introduction to recursive partitioning using the rpart routines.* Technical Report 61. URL http://www. mayo. edu/hsr/techrpt/61. pdf.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with s* (Fourth ed.). New York: Springer. Retrieved from `http://www.stats.ox.ac.uk/pub/MASS4` (ISBN 0-387-95457-0)

West, B. T. (2006). A simple and flexible rating method for predicting success in the ncaa basketball tournament. *Journal of Quantitative Analysis in Sports*, *2*(3).

West, B. T., & Lamsal, M. (2008). A new application of linear modeling in the prediction of college football bowl outcomes and the development of team ratings. *Journal of Quantitative Analysis in Sports*, *4*(3).

Wickham, H. (2016). rvest: Easily harvest (scrape) web pages [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=rvest` (R package version 0.3.2)

Appendix

Table 26 contains an overview of the championships that are used in this research to collect race results data from.

Single-seater championships	
ADAC Formula 4	2015, 2016, 2017
All-Japan Formula Three Championship	2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017
Australian Formula 4 Championship	2015, 2016, 2017
BRDC British Formula 3 Championship	2016, 2017
BRDC Formula 4 Championship	2013, 2014, 2015
British Formula 3 International Series	2011, 2012
British Formula Three Championship	2013, 2014
British Formula Three Championship	2008, 2009, 2010
China Formula 4 Championship	2015, 2016, 2017
Euro Formula Open	2014, 2015, 2016, 2017
Eurocup Formula Renault 2.0	2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017
European F3 Open Championship	2009, 2010, 2011, 2012, 2013
F4 British Championship	2016, 2017
F4 Danish Championship	2017
F4 Japanese Championship	2015, 2016, 2017
F4 Spanish Championship	2016, 2017
FIA Formula 3 European Championship	2012, 2013, 2014, 2015, 2016, 2017
Formula 3 Brasil	2014, 2015, 2016, 2017
Formula 3 Euro Series	2008, 2009, 2010, 2011
Formula 4 South East Asia Championship	2017
Formula 4 UAE Championship	2016, 2017
Formula 4 United States Championship	2016, 2017
Formula Renault 2.0 NEC	2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017
GP3 Series	2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017
Italian F4 Championship	2014, 2015, 2016, 2017
MSA Formula Championship	2015
NACAM Formula 4 Championship	2016, 2017
SMP F4 Championship	2015, 2016, 2017
Kart championships	
CIK-FIA European Championship KF	2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015
CIK-FIA European Championship KZ	2013, 2014, 2015, 2016, 2017
CIK-FIA World Championship KF	2012, 2013, 2014, 2015
CIK-FIA World Championship KZ	2013, 2014, 2015, 2016, 2017
WSK Euro Series KZ1	2010, 2011, 2012, 2013

Table 26: Championships that are used to collect data from.