Northwestern University

Machine Learning-Based Prediction of Formula One Race Outcomes

A Data-Driven Study of Top-10 Finish Prediction Using Historical Race Data

Prepared by:

Sara Alsiyat - Qifan Yang - Boqi Niu

MSDS 422: Practical Machine Learning

Dr. Irene Tsapara

Northwestern University

January 2026

Abstract

Formula One race outcomes are influenced by multiple factors, including driver performance, constructor strength, circuit characteristics, and race conditions. Finishing in the top 10 of a race is particularly important because it determines whether a driver earns championship points and reflects competitive performance. Historical race data provides an opportunity to study these factors and understand patterns in race outcomes. This project examines whether machine learning methods can be used to predict whether a driver finishes in the top 10 of a Formula One race using historical data.

The study uses a publicly available Formula One World Championship dataset from Kaggle, covering races from 1950 to recent seasons. The dataset is provided in multiple relational tables and includes race results, qualifying data, driver and constructor information, race status, and circuit details. The unit of analysis is at the driver–race level, where each record represents a driver's result in a specific race. The dataset contains both numerical attributes, such as finishing position, grid position, points, laps completed, and season indicators, and categorical attributes, including driver, constructor, circuit, country, and race status.

The target variable for this project is a binary indicator representing whether a driver finishes within the top 10 positions in a race. The purpose of the study is to evaluate whether historical race information can be used to support structured prediction of race outcomes. The project is intended to provide a data-driven foundation for analyzing Formula One performance and exploring the potential of machine learning techniques in competitive sports analytics.

Problem Statement

Formula One race results depend on many factors, including driver ability, constructor performance, track characteristics, and race conditions. Although race outcomes are often seen as unpredictable, teams and analysts use historical data to better understand what drives performance. Finishing in the top 10 is especially important because it determines whether a driver earns championship points and reflects overall race success.

This project focuses on predicting whether a driver will finish in the top 10 of a Formula One race using historical race data. Machine learning methods are used to combine information about drivers, constructors, circuits, and recent performance. The goal is to understand whether race outcomes can be explained using measurable factors rather than chance, and whether predictive models can support performance analysis and strategic decision making in Formula One.

Research objective

The main objective of this project is to build and evaluate machine learning models that predict top-10 finishes in Formula One races using historical data. The study examines how factors such as qualifying position, constructor team, driver performance history, track characteristics, and recent race results affect finishing position. The project also explores whether some circuits show more variation in race outcomes, which may suggest more unpredictable races compared to tracks where performance is more consistent. In addition, the research considers related outcomes such as points finishes and Did Not Finish (DNF) events to better understand race reliability and performance risk. Finally, different machine learning models will be compared to assess their accuracy, interpretability, and usefulness for decision-support purposes.

Annotated Bibliography

Race to the Podium: Separating and Conjoining the Car and Driver in F1 Racing – Alsiyat

This paper studies what really drives race outcomes in Formula One by separating the impact of the driver from the impact of the car and constructor. The authors use historical Formula One race results and financial data from multiple seasons, focusing on races between 2012 and 2019. The dataset includes information on drivers, constructors, race finishing positions, qualifying positions, retirements (DNF), team budgets, and driver salaries. The main goal of the study is to understand how much each factor contributes to a driver's finishing position and whether success in Formula One is driven more by individual skill, team resources, or the interaction between both. The paper applies statistical models to race-level data and focuses on measurable outcomes such as finishing position and podium results.

The results show that both driver skill and constructor performance play an important role in determining race outcomes, but their effects are not equal. While driver ability matters, the interaction between the driver and the team explains a large portion of race performance. High-performing drivers tend to outperform their teammates even in similar cars, while strong constructors provide advantages through better technology, strategy, and reliability. The study also shows that race outcomes are not random and that historical performance can be used to explain and anticipate future results.

This paper is highly relevant to our project because it directly supports our goal of predicting whether a driver will finish in the top 10 of a race. From a decision-support perspective, the clear separation between driver and constructor effects justifies including both driver-level and team-level features in our machine learning models. In my professional experience working with national-level data and KPIs, separating contributing factors is critical for building reliable and explainable models. The

paper's findings also support using interaction features, such as driver–constructor combinations, which aligns well with our planned feature engineering using the Kaggle Formula One dataset.

NeuralAC: Learning Cooperation and Competition Effects for Match Outcome Prediction - Alsiyat

This paper introduces NeuralAC, a machine learning framework designed to predict match outcomes by modeling both cooperation and competition effects between participants. The authors evaluate the model using historical match outcome data from competitive team-based environments, where each observation represents a match involving multiple participants working together against opponents. The dataset includes team compositions, participant identities, and match results, allowing the model to learn how interactions influence outcomes rather than relying only on individual performance metrics. NeuralAC uses neural networks to capture these interaction patterns and compare them against traditional prediction models.

The key contribution of this paper is showing that modeling interaction effects leads to more accurate predictions than treating participants independently. The results demonstrate that cooperation within a team and competition between teams both strongly influence outcomes. The experiments show that NeuralAC performs better in complex competitive settings where outcomes depend on coordination, shared resources, and strategic interactions. This highlights the limitation of models that rely only on individual features.

This paper is important for our project because Formula One race outcomes also depend on interaction effects, especially between the driver and the constructor. While Formula One appears to be an individual sport, performance depends heavily on teamwork, car reliability, race strategy, and

coordination. From a decision-support perspective, this paper supports going beyond driver-only features and incorporating interaction features in our models. In applied analytics work, capturing how factors work together often produces more useful insights than analyzing them in isolation. NeuralAC provides strong methodological support for experimenting with models that can capture complex relationships when predicting top-10 finishes using historical Formula One data.

A Data-Driven Analysis of Formula 1 Car Races Outcome - Yang

This article gives us a statistical analysis to help understand the most influential factors that can impact the outcome of Formula 1 races. The outcomes are measured as total championship points scored by a driver over the period of a season. The main motivation is that F1 teams collect vast amounts of data, but yet few studies or analysis systematically analyzes which variables really affect race results. In the study, the researchers compiled race data from 2015 to 2019 by web-scraping and processed the data into a dataset with 21 features such as pit frequency, tyre usage percentages, laps led, penalties, race positions, starting positions, accidents, and driver completion rates. First, correlation analysis was conducted and it turned out that there are strong relationships among position-related variables and between tyre usage and penalties. This suggests that there are complex dependencies among factors. Due to the high feature interdependence, the authors used Principal Component Analysis (PCA) to reduce the dimensionality. The first four principal components can capture around 70% of the total variance. This indicates that the major patterns can be represented in a lower-dimensional space without significant loss of information.

Next, a linear regression model is used to predict total points scored by drivers (based on the reduced feature set). Key findings include that finishing more races correlates strongly with higher points, while tyre usage of certain compounds such as soft, medium, hard also significantly affects performance. Additionally, a better average starting position or pole position is associated with more points. This illustrates the importance of qualifying. The model achieved an R-square of 99%, which suggests a strong explanatory power on this dataset.

The authors conclude systematic statistical analyses can discover meaningful insights into F1 race performance.

Advanced Machine Learning Approaches for Formula 1 Race Performance Prediction: A Comprehensive Analysis of Championship Point Forecasting - Yang

This study presents a machine learning framework designed to predict Formula 1 race performance and championship points. It used a dataset which spans 74 years of F1 history from 1950 to 2024. This includes 589081 individual lap times from 1125 total races. The authors' goal was to overcome limitations in prior works by using complex feature engineering, algorithm comparisons, and rigorous validation methods which are specially geared towards complex motorsport analysis.

Their approach begins with building meaningful features from qualifying race data, lap time data, circuit characteristics, and temporal dynamics across different eras of racing from 1950 to 2024. These engineered features capture strategic and performance factors such as grid position effects, lap-to-lap variations, and historical performance patterns across different circuits. Various machine learning models

are evaluated, including traditional regression, ensemble methods, and gradient boosting techniques. It also comes with hyperparameter tuning and cross-validation to ensure robust generalization.

Among the tested algorithms for example gradient boosting such as XGBoost consistently outperforms others, yielding exceptional predictive accuracy with an R-square score of 0.999, RMSE of 0.197, and MAE of 0.125 on forecasting championship points. Feature importance analysis indicates that race position is the dominant predictor, which contributes around 75.8 % to model performance, with seasonal variations accounting for another 23.8 %. Cross-validation results demonstrate strong generalization (mean R-square roughly equals to 0.993 plus or minus 0.013).

The paper argues that its findings not only advance predictive modeling performance in motorsport but also hold some practical implications for F1 teams, broadcasters, and strategy analysts. It also offers insights into how historical performance and context can influence race outcomes.

The Use of Machine Learning in Predicting Formula 1 Race Outcomes - Niu

This paper examines the application of machine learning techniques to predicting Formula 1 race outcomes, with a particular focus on driver finishing positions and constructor championship points. Using historical Formula 1 data from the 2010 to 2023 seasons, the study develops a structured predictive modeling pipeline based on TabNet, a deep learning architecture specifically designed for tabular data. TabNet is chosen for its strong predictive performance as well as its built-in interpretability through feature attention mechanisms.

Two separate models are constructed: a driver-level model that predicts individual finishing positions and a constructor-level model that forecasts total team points per race. The models rely

exclusively on pre-race features such as grid position, number of laps, constructor affiliation, and derived indicators like overtaking potential. Data preprocessing includes cleaning incomplete records, encoding categorical variables, feature normalization, and chronological train–test splitting to prevent data leakage. Hyperparameter tuning is performed using Optuna to optimize model performance.

Empirical results demonstrate that the driver model achieves strong predictive accuracy (R^2 = 0.75, RMSE = 2.87), outperforming the constructor model, which exhibits slightly higher variability due to team-level complexity (R^2 = 0.71, RMSE = 3.92). Visualization and residual analyses indicate low systematic bias and good alignment between predicted and actual outcomes. The study concludes that interpretable deep learning models like TabNet can effectively predict Formula 1 race outcomes using structured pre-race data. Limitations include the absence of real-time race variables such as weather, safety cars, and telemetry, which are identified as promising directions for future research.

Predicting Formula 1 Race Outcomes: A Machine Learning Approach - Niu

This paper investigates the use of machine learning techniques to predict Formula 1 lap times and race outcomes using historical race data from 2014 to 2023. The primary objective is to assess how accurately lap times for individual drivers can be forecasted and how these predictions translate into overall race results. The study emphasizes the importance of modeling temporal dependencies in racing data, given that lap performance is influenced by prior laps, race conditions, and driver history.

The author constructs a large-scale dataset comprising over 214,000 laps across 203 races and 55 drivers, incorporating core performance metrics, race conditions (such as pit stops and safety cars), and driver- and team-specific indicators. Several baseline models—including linear regression, decision

trees, and random forests—are first evaluated to establish reference performance levels. While random forests outperform simpler models, their prediction errors remain large relative to the typical time gaps between drivers, limiting their usefulness for accurate race outcome prediction.

To better capture sequential patterns, the study focuses on Long Short-Term Memory (LSTM) networks. Multiple LSTM architectures are developed and refined through increased training epochs, dropout regularization, custom feature scaling, and a novel composite loss function that combines lap time error, positional accuracy, and historical driver performance. Experimental results demonstrate that minimizing lap time error alone does not necessarily lead to better race outcome predictions, highlighting the importance of incorporating relative and contextual performance measures.

Evaluation on selected races from the 2023 and 2024 seasons—particularly the Abu Dhabi and Bahrain Grands Prix—shows that later LSTM models substantially outperform baseline approaches in predicting podium finishers, top-five placements, and race winners, especially in races with fewer unexpected incidents. The paper concludes that LSTM-based models, when properly designed and evaluated, offer significant promise for predictive analytics in Formula 1. Future work is proposed to explore transformer-based models and to extend prediction scope to include pit stop strategies and safety car events, aiming for a more comprehensive race forecasting system.

NBA Winner Prediction: A Hybrid Framework Incorporating Internal and External Factors – Alsiyat

This paper studies how machine learning can be used to predict the outcome of NBA games by combining both internal and external factors. The authors use historical NBA regular season data from the 2012–13 season to the 2021–22 season, collected from professional sports databases. The dataset

includes game-level information such as team performance statistics, win–loss outcomes, home and away status, Elo ratings, and contextual factors like home court advantage and player tiredness due to back-to-back games. The main objective of the study is to improve prediction accuracy by incorporating contextual and situational features in addition to traditional performance metrics. The authors apply supervised machine learning models and evaluate their performance using cross-validation.

The results show that combining internal team performance indicators with external contextual factors leads to better prediction accuracy compared to models that rely only on basic statistics. Features such as differences in Elo ratings, recent team performance, home court advantage, and tiredness were found to be important predictors of game outcomes. Among the tested models, ensemble-based approaches such as random forest achieved the strongest performance. The study demonstrates that sports outcomes are influenced by both historical performance and situational conditions, and that machine learning models can capture these relationships effectively.

This paper is relevant to our project because it provides a clear example of how combining multiple types of features improves outcome prediction in competitive sports. From a decision-support perspective, the paper supports our approach of integrating driver-level, constructor-level, and contextual race information when predicting top-10 finishes in Formula One. Similar to how NBA outcomes depend on both team strength and external conditions, Formula One race results depend on driver skill, car performance, circuit characteristics, and race context. In applied analytics work, incorporating both internal and external factors leads to more reliable and explainable models. The methodology used in this paper helps justify our feature selection strategy and supports the use of supervised machine learning models for structured performance prediction using historical Formula One data.

Bayesian analysis of Formula One race results: disentangling driver skill and constructor advantage - Yang

A very interesting and valuable question is posed through the article regarding analytics in Formula 1 racing by determining how driver skill and constructor advantage affect performance. This concept is explored through Bayesian analysis within the work of Erik-Jan van Kesteren and Tom Bergkamp. The beauty of sports analytics, in a very simple and instinctive way, is the ability to rank competition participants by their individual ability; however, in high-speed racing and high-stakes competitions (i.e., Formula 1) there are many influences and other types of factors that come into play and affect the outcome of the competition very differently than in many other forms of sport. The probing and inquisitive thinking of the millions of fans and devotees of this sport, regarding questions like "What is the effect of the driver's skill versus the constructor?" remain unanswered.

In order to accomplish this, the authors have created a new Bayesian multilevel (hierarchical) model for rank-ordered logit regression that directly uses data from race finishing positions during the hybrid era of Formula One (2014–2021). The current model improves upon previous efforts by not losing information from point-based systems; it also uses the performance differences of teammates and driver movements between teams to differentiate between driver and constructor effects.

The model computes an entire finish order (rather than aggregate points) allowing for greater details regarding competitive dynamics.

Through this Bayesian framework, the authors are able to measure the skill of each driver and the advantage of each constructor (in log-odds terms) in relation to their competitors (analogous to Elo ratings in chess) with credible intervals to measure uncertainty. The findings show that team performance (constructor) has a large impact on the outcomes of races; approximately 88% of the variance in results

is associated with the car, and less than half of the remaining variance can be attributed to driver skill and/or other factors. Also, in terms of driver performance during this time period, the top two drivers were Lewis Hamilton and Max Verstappen, while the top three constructors were Mercedes, Ferrari and Red Bull.

This is a good study that explains exactly how the proposed method can be used to create counterfactual scenarios (such as what if a driver had been in a different type of car) that can be used to rank each individual driver and constructor independently while providing a statistical basis for all of this. The authors conclude that this methodology advances quantitative analytical rigor for measuring the performance of Formula One drivers but that it may extend beyond Formula One to other sporting and competitive settings where the interaction between multiple variables in performance measurement is relevant.

Learning to Identify Top Elo Ratings: A Dueling Bandits Approach - Niu

This paper addresses the problem of efficiently identifying top-performing players under the Elo rating system by reducing the number of matches required to accurately estimate player strength. Traditional Elo-based evaluation relies on repeated random or predefined match-ups, which can be inefficient, especially when the goal is to quickly identify the strongest players among a large pool of competitors. To overcome this limitation, the authors formulate the match scheduling problem as a dueling bandits problem and propose an adaptive, data-driven framework for selecting informative match-ups.

The paper introduces two online algorithms, MaxIn-Elo and MaxIn-mElo, which actively choose which pairs of players should compete at each step. Instead of scheduling matches uniformly at random, the algorithms focus on players whose Elo ratings are uncertain but potentially high-performing. This is achieved by maintaining confidence bounds on Elo estimates and prioritizing match-ups that maximize expected information gain. Elo ratings are updated online using stochastic gradient descent, allowing the system to scale efficiently while using constant memory per iteration.

From a theoretical perspective, the authors show that MaxIn-Elo achieves sublinear cumulative regret, meaning that the algorithm becomes increasingly efficient over time compared to random or naive scheduling strategies. The framework is also extended to multidimensional Elo (mElo), which allows the model to capture intransitive competitive relationships, such as scenarios where no single player consistently dominates all others. This extension is particularly important in competitive environments where performance depends on interaction effects rather than a simple global ranking.

The proposed methods are evaluated through extensive experiments on both synthetic data and real-world competitive games. Results show that MaxIn-Elo and MaxIn-mElo converge significantly faster and produce more accurate rankings than baseline approaches, including random sampling and existing dueling bandit methods. In both transitive and intransitive settings, the algorithms demonstrate superior performance in identifying top players with fewer comparisons.

Overall, the paper demonstrates that combining adaptive match scheduling with online Elo updates provides a more efficient and scalable approach to ranking competitors. While the study focuses primarily on identifying the best player and assumes stationary skill levels, the framework offers valuable insights for ranking and outcome prediction tasks in complex competitive systems.

References

Vopani. “Formula 1 World Championship (1950 - 2024).” Kaggle, January 29, 2025. https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020.

Rockerbie, Duane W., and Stephen T. Easton. “Race to the Podium: Separating and Conjoining the Car and Driver in F1 Racing.” Applied Economics 54, no. 54 (July 5, 2022): 6272–85. https://doi.org/10.1080/00036846.2022.2083068.

Gu, Yin, Qi Liu, Kai Zhang, Zhenya Huang, Runze Wu, and Jianrong Tao. “Neuralac: Learning Cooperation and Competition Effects for Match Outcome Prediction.” Proceedings of the AAAI Conference on Artificial Intelligence35, no. 5 (May 18, 2021): 4072–80. https://doi.org/10.1609/aaai.v35i5.16528.

Patil, Ankur, Nishtha Jain, Rahul Agrahari, Murhaf Hossari, Fabrizio Orlandi, and Soumyabrata Dev. “A Data-Driven Analysis of Formula 1 Car Races Outcome.” Communications in Computer and Information Science, 2023, 134–46. https://doi.org/10.1007/978-3-031-26438-2_11.

Bansal, Aayam. Advanced Machine Learning Approaches for Formula 1 Race Performance Prediction: A Comprehensive Analysis of Championship Point Forecasting, 2025. https://doi.org/10.13140/RG.2.2.20910.01607.

Urdhwareshe, A. 2025 "The Use of Machine Learning in Predicting Formula 1 Race Outcomes" Preprints. https://doi.org/10.20944/preprints202504.1471.v1

Jafri, Ali. *Predicting Formula 1 Race Outcomes: A Machine Learning Approach*. Capstone project, Fall 2024.

https://aliabdullahjafri.com/static/media/Ali_Jafri_CapstoneProject1_Fall2024.c7244022875d46bec5d9.pdf.

Zheng, Xi. “NBA Winner Prediction: A Hybrid Framework Incorporating Internal and External Factors.” *2022 4th International Conference on Big Data Engineering*, May 26, 2022, 71–80. https://doi.org/10.1145/3538950.3538960.

Kesteren, Erik-Jan van, and Tom Bergkamp. “Bayesian Analysis of Formula One Race Results: Disentangling Driver Skill and Constructor Advantage.” Journal of Quantitative Analysis in Sports 19, no. 4 (July 25, 2023): 273–93. https://doi.org/10.1515/jqas-2022-0021.

NeuralAC: Learning Cooperation and Competition Effects for Match Outcome Prediction

Yin Gu[1], Qi Liu[1]*, Kai Zhang[1], Zhenya Huang[1], Runze Wu[2], Jianrong Tao[2]

[1] Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science & School of Computer Science and Technology, University of Science and Technology of China

[2] Fuxi AI Lab, NetEase Inc., Hangzhou, China

{gy128, kkzhang0808}@mail.ustc.edu.cn, {qiliuql, huangzhy}@ustc.edu.cn,

{wurunze1, hztaojianrong}@corp.netease.com

Abstract

Match outcome prediction in group comparison setting is a challenging but important task. Existing works mainly focus on learning individual effects or mining limited interactions between teammates, which is not sufficient for capturing complex interactions between teammates as well as between opponents. Besides, the importance of interacting with different characters is still largely underexplored. To this end, we propose a novel ***Neural Attentional Cooperation-competition model*** (***NeuralAC***), which incorporates weighted-cooperation effects (i.e., intra-team interactions) and weighted-competition effects (i.e., inter-team interactions) for predicting match outcomes. Specifically, we first project individuals to latent vectors and learn complex interactions through deep neural networks. Then, we design two novel attention-based mechanisms to capture the importance of intra-team and inter-team interactions, which enhance NeuralAC with both accuracy and interpretability. Furthermore, we demonstrate NeuralAC can generalize several previous works. To evaluate the performances of NeuralAC, we conduct extensive experiments on four E-sports datasets. The experimental results clearly verify the effectiveness of NeuralAC compared with several state-of-the-art methods.

Introduction

Group comparison, usually involving two teams competing with each other (e.g., Figure 1), is ubiquitous in sports and online games, such as football, *Dota2*, and *League of Legends*. In the last decade, the popularity of online competitive games has exploded and there are more than 800 million online game players. A large number of players create great commercial value coupled with some technical challenges. One of the crucial problems, i.e., match outcome prediction, has attracted considerable research attention since it plays a key role in creating fair matches for players and increasing the teams' probability of winning (Chen et al. 2018).

In the literature, many existing methods in group comparison (Herbrich, Minka, and Graepel 2007; Huang, Lin, and Weng 2008) focus on learning individual effects from outcomes of group comparisons. Despite the popularity of these methods, they omit interplays between players within a team. In their assumption, team members are independent of each other, that is, a team's ability is modeled as the sum of the team members' score. To tackle this limitation, neural network-based methods (Delalleau et al. 2012; Gong et al. 2020) were proposed to capture the intra-team interactions. However, they focus on obtaining the team representation by aggregating single team member's representations, which preserves little low-level information, and thus it is hard to evaluate individuals' contributions to the teamwork. Meanwhile, factorization machines (FM) (Rendle 2010) was adopted to group comparison (Li et al. 2018a), where the cooperation effect (i.e., intra-team interaction) was modeled as the inner product of two latent vectors. Though low-level information was preserved, they can not model non-linear interactions due to the limitation of FM method.

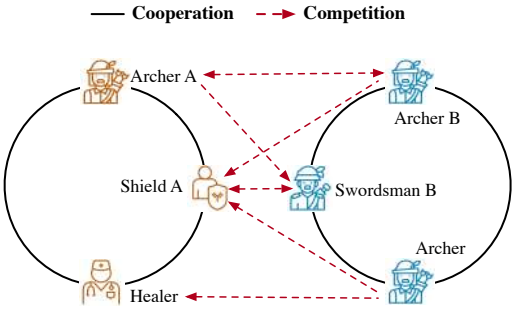


Figure 1: An example of group comparisons.

Indeed, both cooperation and competition are very common in human society (Bengtsson and Kock 1999; Bar-Yam 2003; Tauer and Harackiewicz 2004; MacRae 2018), which can be highly complex. For example, as shown in Figure 1, two teams fight each other in group comparison (e.g., battlefield), which involves multiple interactions, including intra-team interactions (e.g., the shield soldier A protects teammates, the healer cures teammates), and inter-team interactions (e.g., the archer A shoots the swordsman B, the shield soldier resists the swordsman's attack). Everyone on the battle has different strengths and weaknesses, making them perform differently when against different opponents. Meanwhile, teammates could complement each other through cooperation, which makes the group comparison highly intri-

*Corresponding Author.

cate. To make the prediction accurate, it's necessary to incorporate comprehensive interactions. Nevertheless, how to model such cooperation and competition effects simultaneously remains a challenge.

Another limitation of existing approaches is that they do not consider the importance of interactions. Considering an ancient war where two armies fight each other, soldiers focus on finding opportunities to kill enemy generals. Meanwhile, soldiers try to protect generals on their side. It's clear that generals play a key role in wars, and interacting with generals has a larger influence on the outcome of the war. Therefore, in this group comparison scenario, generals should receive more attention. In other words, interactions with different characters have different attention scores, as they contribute differently to the match outcome. Thus, modeling inter-team and intra-team attention distributions is a nontrivial task and yet remains a challenge.

To address the above challenges, in this paper, we propose a novel ***Neural Attentional Cooperation-competition model*** (***NeuralAC***), which incorporates weighted-cooperation effects and weighted-competition effects for predicting match outcomes. Different from previous approaches, we choose the element-wise product of two latent vectors as the input of deep neural networks (DNNs) to get the corresponding score, which greatly facilitates deep layers to learn meaningful second-order interactions, while still preserving its interpretability. First, by deploying DNNs on both intra-team and inter-team interactions, we get pairwise cooperation scores and pairwise competition scores. Then, we design two attention mechanisms to capture the importance of intra-team and inter-team interactions, which enhance NeuralAC with both accuracy and interpretability. Furthermore, we demonstrate NeuralAC is general and expressive, which can generalize several previous works. The main contributions of this work are as follows:

- We consider both intra-team interactions and inter-team interactions, and we propose to model comprehensive interactions with neural networks for learning complex cooperation and competition effects.
- We further propose two attention mechanisms to enhance NeuralAC, which provide strong interpretability about the importances of interactions.
- Extensive experiments on four real E-sports datasets show the effectiveness of NeuralAC. The code and datasets are avaliable at https://github.com/bigdata-ustc/NAC.

Related Work

Group Comparison

Many existing works (Herbrich, Minka, and Graepel 2007; Huang, Lin, and Weng 2008) in this area focus on learning individual effects from group comparison. They assume the player's performance is independent of teammates, and the ability of the team is represented as the summation of the team members' scores. This assumption may not hold true in the real world, because some players may perform well when they team-up together. To address this limitation, some methods (DeLong et al. 2011; Semenov et al. 2016; Li et al. 2018a) are proposed to model the cooperation effects in the team composition. For instance, Li et al.(2018a) exploited factorization machine to model interplay between teammates. Their methods may not be expressive enough due to intra-team interactions is modeled in a linear way. Deep learning is also adopted (Gong et al. 2020; Delalleau et al. 2012) to capture intra-team interactions. Gong et al.(2020) also proposed a novel technique of learning the representations of individuals from relation graphs. However, these works mainly utilize DNNs for aggregating players' representations to obtain team representations. Despite non-linear interactions is modeled, their methods capture limited information at the low level. Besides, due to the inherent traits of DNNs, these methods lack interpretability and it's hard to assess individuals' contributions to team works.

The existing works either focus on learning individual effects or modeling limited cooperation effects. Besides, competition effects and the importance of interactions are still largely under-explored. Meanwhile, some methods (Delalleau et al. 2012; Minka, Cleven, and Zaykov 2018; Gong et al. 2020) utilized in-game features to get more accurate predictions. However, those features are usually designed by experts in the domain, hence case-specific. We focus on a more general task with no domain knowledge required. Therefore, we don't utilize any in-game features.

Cooperation and Competition

Cooperation and competition are important factors in other fields, which are widely studied. For example, Dai et al. (2020) predicted cooperation and competition relationships among companies in a company relation network. Usmani et al. (2020) analyzed the competitiveness of commercial products in the market. Some works (Lowe et al. 2017; Wray, Kumar, and Zilberstein 2018) explored to model decisions making process in the multi-agent cooperative-competitive environment. Although cooperation and competition have been studied in other fields, very little work in group comparison has fully explored the impact of cooperation and competition effects.

NeuralAC Model

In this section, we first formally introduce match outcome prediction task. Then, we give an overview of NeuralAC. After that, we details basic NeuralAC and attention mechanisms. Finally, we demonstrate the generality of NeuralAC.

Problem Definition

Suppose there are n individuals $\{1, 2, ..., n\}$, M observable matches. Each match involves two teams T_A and T_B, each of them is a subset of $\{1, 2, ..., n\}$, and the match outcomes of the M matches is denoted as $\{y_1, y_2, ..., y_M\}$. In this paper, we focus on the problem of binary match outcome prediction, each match outcome is either win or lose. We assume that there is no draw. Let $y_m = 1$ if T_A beat T_B in a match $m \in [1, M]$, otherwise $y_m = 0$. Given a match between T_A and T_B, our goal is to predict the match outcome $\hat{y} \in [0, 1]$.

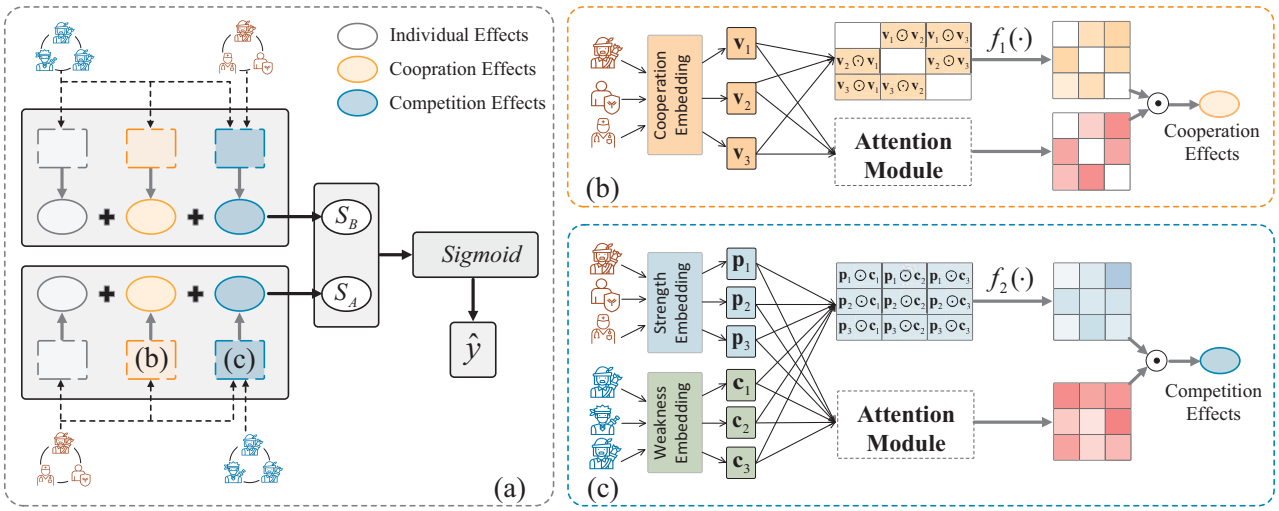


Figure 2: NeauralAC model architecture. (a) shows the overview of the model; (b) is cooperation effects part; (c) is competition effects part. Note that, for clarity purpose, we omit the individual effects part in this figure.

Model Overview

In this paper, we focus on group comparison. We assume each team has a score indicating team's ability. Inspired by Blade-Chest (Chen and Joachims 2016), we formulate the probability of team T_A defeating team T_B as:

$$\begin{aligned} P(A\ beats\ B) &= \frac{\exp(S_A)}{\exp(S_A)+\exp(S_B)}, \\ &= \frac{1}{1+\exp(-(S_A-S_B))}, \\ &= \sigma(\Delta(A,B)), \end{aligned} \tag{1}$$

where S_A is T_A's score that represents the overall ability of the team, σ is the sigmoid function. $\Delta(A,B)$ denotes the edge that T_A have when match up against T_B. When $\Delta(A,B) \to 0$, both teams have the equal odds to win. When $\Delta(A,B) \to +\infty$, $P(A$ beats $B) \to 1$, B almost has no chance to defeat A, and vice versa.

As mentioned above, there are multiple complex interactions in group comparisons (e.g., cooperation between teammates and competition between opponents). Generally, if team members get high individual ability or two members cooperate well, the overall ability of the team can be improved. Besides, if a player in T_A has more advantages when competing with his opponents, then the overall ability of T_A can be further improved. Therefore, in NeuralAC, the overall ability of the team consists of three parts: individual effects, cooperation effects and competition effects. Take T_A versus T_B as an example, we formulate T_A's score as:

$$S_A = \sum_{i \in T_A} w_i + F_{\text{coop}}(T_A) + F_{\text{comp}}(T_A, T_B), \tag{2}$$

where w_i indicates i's individual ability, which is model parameter. The first term of S_A models individual effects. The second term $F_{\text{coop}}(T_A)$ and the third term $F_{\text{comp}}(T_A, T_B)$ models cooperation effects and competition effects, respectively. Figure 2 shows the framework and two main components of NeuralAC.

Basic NeuralAC

In this subsection, we illustrate the detail of NeuralAC without two attention mechanisms (e.g., cooperation effects part, competition effects part).

Cooperation Effects. Since everyone has different cooperation characteristics, the cooperation effect usually differs when working with different teammates. Inspired by (Li et al. 2018a; Gong et al. 2020), in NeuralAC, we assume each individual i has an embedding vector $\mathbf{v}_i \in \mathbb{R}^k$, namely cooperation vector, representing his cooperation characteristics. The cooperation effects $F_{\text{coop}}(T_A)$ is formulated as:

$$F_{\text{coop}}(T_A) = \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} f_1(\mathbf{v}_i \odot \mathbf{v}_j), \tag{3}$$

where $\odot$ denotes the element-wise product, $\mathbf{v}_i$ and $\mathbf{v}_j$ are learnable parameters, f_1 refers to the MLP with non-linear activation function, which are capable of learning higher-order and non-linear interactions between teammtes. The output of $f_1(\mathbf{v}_i \odot \mathbf{v}_j)$ is a scalar value, which is the cooperation score between i and j.

Competition Effects. When a player attack another, the competition result depends on the offensive's strength and the defensive's weakness, and vice versa. Inspired by Blade-Chest, in NeuralAC, each individual i has two distinctive embedding vectors $\mathbf{p}_i \in \mathbb{R}^k$, $\mathbf{c}_i \in \mathbb{R}^k$, namely strength vector and weakness vector, respectively. To simplify the setting, we assume $\mathbf{v}_i$, $\mathbf{p}_i$, $\mathbf{c}_i$ share the same size k. Then, the competition effects $F_{\text{comp}}(T_A, T_B)$ is formulated as:

$$F_{\text{comp}}(T_A, T_B) = \sum_{i \in T_A} \sum_{j \in T_B} f_2(\mathbf{p}_i \odot \mathbf{c}_j), \tag{4}$$

where $\odot$ denotes the element-wise product, $\mathbf{p}_j$ and $\mathbf{c}_i$ are learnable parameters, f_2 refers to a MLP with non-linear activation function, which can model non-linear interactions between opponents. The output of $f_2(\mathbf{p}_i \odot \mathbf{c}_j)$ is a

scalar value, indicating a competition score when i compete against j. To summarize, we give the formulation of S_A as:

$$S_A = \sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} f_1(\mathbf{v}_i \odot \mathbf{v}_j) + \sum_{i \in T_A} \sum_{j \in T_B} f_2(\mathbf{p}_i \odot \mathbf{c}_j). \tag{5}$$

DNNs Components. In our setting, f_1 and f_2 share the same network structure. Here, we elaborate the design of f_2. To simplify the description, we denote $\mathbf{p}_i \odot \mathbf{c}_j$ as $\mathbf{x}$, and then feed $\mathbf{x}$ to the MLP. Similar to NFM (He and Chua 2017), the process can be formulated as:

$$\begin{aligned} \mathbf{z}_1 &= \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1), \\ \mathbf{z}_2 &= \sigma(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2), \\ &\dots \qquad\qquad \dots \\ \mathbf{z}_L &= \sigma(\mathbf{W}_L \mathbf{z}_{L-1} + \mathbf{b}_L), \end{aligned} \tag{6}$$

where L indicates the number of hidden layers, $\mathbf{W}_L$, $\mathbf{b}_L$ denote the weight matrix and bias for the L-th layer. $\sigma(\cdot)$ is $ReLU$ activation function.

To ensure the interpretability of NeuralAC (e.g. it does not make sense when cooperation and competition scores are negative), we set the activation function of the output layer to be $ReLU$.:

$$O_{ij} = ReLU(\mathbf{W}_o \mathbf{z}_L + \mathbf{b}_o), \tag{7}$$

where O_{ij} is the competition score when i against j.

NeuralAC

In this subsection, we show how to enhance basic NeuralAC with two attention mechanisms. Attention mechanisms have been widely used in many tasks, such as computer vision (Chen et al. 2017; Liu et al. 2018), natural language processing (Zhang et al. 2019), and recommendation system (Xiao et al. 2017; Li et al. 2018b). In a team competition case, we often pay more attention to the key person in our team. Similarly, we usually focus on the key person in the opponent team and look for a chance to defeat him. Since cooperating or competing with the key person has a greater influence on the match outcome, not all interactions should share the same weight as they contribute differently to the final game outcome. Motivated by this intuition, we propose to deploy the attention modules on cooperation effects and competition effects as:

$$S_A = \sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, j \neq i} a_{ij}^{\text{coop}} f_1(\mathbf{v}_i \odot \mathbf{v}_j) + \sum_{i \in T_A} \sum_{j \in T_B} a_{ij}^{\text{comp}} f_2(\mathbf{p}_i \odot \mathbf{c}_j), \tag{8}$$

where a_{ij}^{coop}, a_{ij}^{comp} is intra-team and inter-team attention score respectively, which can be interpreted as the importance of the interaction in contributing to the game outcome.

Attention Components. To make attention modules generalized to unseen pairs and asymmetry (e.g., i's attention to j is usually different from j's attention to i), we formulate them as follows:

$$\begin{aligned} r_{ij}^{\text{coop}} &= \mathbf{v}_i^T \mathbf{W}_{\text{coop}} \mathbf{v}_j, \\ a_{ij}^{\text{coop}} &= \frac{\exp(r_{ij}^{\text{coop}})}{\sum_{j \in T_A, j \neq i} \exp(r_{ij}^{\text{coop}})}, \end{aligned} \tag{9}$$

$$\begin{aligned} r_{ij}^{\text{comp}} &= \mathbf{p}_i^T \mathbf{W}_{\text{comp}} \mathbf{c}_j, \\ a_{ij}^{\text{comp}} &= \frac{\exp(r_{ij}^{\text{comp}})}{\sum_{j \in T_B} \exp(r_{ij}^{\text{comp}})}, \end{aligned} \tag{10}$$

where $\mathbf{W}_{\text{coop}} \in \mathbb{R}^{k \times k}$, $\mathbf{W}_{\text{comp}} \in \mathbb{R}^{k \times k}$ are learnable model parameters , r_{ij}^{coop} and r_{ij}^{comp} denote attention values. The inputs to r_{ij}^{coop} are two teammates' cooperation vectors (e.g., $\mathbf{v}_i$ and $\mathbf{v}_j$), and the inputs to r_{ij}^{comp} are one's strength vector (e.g., $\mathbf{p}_i$) and his opponent's weakness vector (e.g., $\mathbf{c}_i$). The higher the r_{ij}^{coop}, the more attention j will receive from his teammate i. The higher the r_{ij}^{comp}, the more attention j will receive from his opponent i.

Training Strategy

Given M observed matches, let y_i denote the i-th match outcome, $\hat{y}_i$ denote corresponding prediction (i.e., $P(A\ beats\ B)$). The loss function is cross entropy between model output $\hat{y}$ and true label y:

$$\mathcal{L} = -\sum_{i=1}^{M} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)). \tag{11}$$

In this way, we can learn NeuralAC by directly minimizing the loss function $\mathcal{L}$.

Generality of NeuralAC

In this subsection, we demonstrate the generality of NeuralAC, and pervious works can be seen as special cases of NeuralAC. To be specific, we first simplify NeuralAC by removing attention modules and non-linear activation function, then set hidden layer number to 0. In this way, T_A's score is formulated as:

$$S_A = \sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} \mathbf{h}_1^T (\mathbf{v}_i \odot \mathbf{v}_j) + \sum_{i \in T_A} \sum_{j \in T_B} \mathbf{h}_2^T (\mathbf{p}_i \odot \mathbf{c}_j), \tag{12}$$

where vector $\mathbf{h}_1 \in \mathbb{R}^k$, $\mathbf{h}_2 \in \mathbb{R}^k$ denotes neuron weights of the output larger.

Generalized Bradly-Terry. Generalized BT (Huang, Lin, and Weng 2008) consider individual effect, while neglecting cooperation effect and competition effect. By fixing $\mathbf{h}_1$ and $\mathbf{h}_2$ to constant zero vectors, we can get the Generalized BT model, where S_A is defined as:

$$S_A = \sum_{i \in T_A} w_i. \tag{13}$$

Factorization Machine. FM (Li et al. 2018a) models pairwise intra-team interactions by inner product of two latent vectors, the team score can be represented as:

$$S_A = \sum_{i \in T_A} w_i + \sum_{i \in T_A} \sum_{j \in T_A, i \neq j} \mathbf{v}_i^T \mathbf{v}_j. \tag{14}$$

By forcing $\mathbf{h}_2$ to be a constant zero vector, and fixing $\mathbf{h}_1$ to be constant one vector, we can get the FM model exactly.

Blade-Chest-Inner. Blade-Chest (Chen and Joachims 2016) model each player i with an absolute ability value w_i, strength vector $\mathbf{p}_i$, and weakness vector $\mathbf{c}_i$. Blade-Chest consider interaction between a opponent, but it is designed for 1v1 circumstance. In Blade-Chest-Inner model, take player a versus player b as an example, a's score is modeled as:

$$S_a = w_a + \mathbf{p}_a^T \mathbf{c}_b. \tag{15}$$

By setting the team size of both sides to 1 and let $\mathbf{h}_2$ to be constant one vector, our model can be reduced to the following formula, which is the same with Blade-Chest model:

$$S_A = \sum_{i \in \{a\}} w_i + 0 + \sum_{i \in \{a\}} \sum_{j \in \{b\}} \mathbf{p}_i^T \mathbf{c}_j. \tag{16}$$

Experiments

Dataset Description

Online games are an ideal testbed and can provide a lot of group comparison data. We use four E-sports datasets to evaluate the utility of our model. The basic statistics of all the datasets are summarized in Table 1.

Dota2 is a famous Multiplayer Online Battle Arena (MOBA) game. In each game, two teams fight each other on the map. Each player controls a different virtual character named hero throughout the whole game. We downloaded ranked matches from yasp.co[1] and Varena[2], which were played in the years of 2015 and 2018 respectively.

League of Legends (LOL) shares a similar game pattern as Dota2. We crawled the recent matches from RiotGame[3]. For *LOL* dataset, we investigate a special game mode, where players are forced to fight on one single line, instead of three lines in Dota2.

We filter out matches that played less than 15 minutes for Dota2 and 8 minutes for LoL. Due to the existence of matchmaking systems, players on both sides have relatively close skills. Therefore, we treat each hero as an individual.

Teamfight Tactics (TFT) is a round-based strategy game that players compete against seven other opponents by constructing and optimizing team compositions to be the last one standing. A Team is composed of heroes selected by the player, each hero has different synergies and equipment. Different from MOBA games, players do not have control of the deployed heroes during the combat time. In each round, the player will fight against a random opponents. We crawl the ranked TFT match records via RiotGame API. We sample group comparisons according to players' last survival round.

[1] https://github.com/odota/core/wiki/JSON-Data-Dump

[2] https://open.varena.com/documentation/dota2/

[3] https://developer.riotgames.com/apis#match-v4

Dataset	Matches	#Heroes	Mode
Dota2015	800,000	110	5v5
Dota2018	580,270	116	5v5
LoL	754,700	148	5v5
TFT	800,000	188	N1vN2

Table 1: Statistics of the datasets.

Baseline Methods

- Logistic Regression (LR) (Ng and Jordan 2002): A linear classifier with L2 regularization. We use the same data input format as Semenov et al. (2016).
- Generalized Bradly-Terry (BT) (Huang, Lin, and Weng 2008): Another linear model, which consider only individual effects.
- TrueSkill (Herbrich, Minka, and Graepel 2007): An algorithm based on probability graph, which is widely used in online games for matchmaking.
- LightGBM (LGB) (Ke et al. 2017) : A highly efficient implementation of GBDT, which achieve state of the art performance in many data science competitions.
- HOI (Li et al. 2018a): A factorization machines (FM) (Rendle 2010) based model that takes pair-wise interactions of teammates into account.
- OptMatch (Gong et al. 2020): A method based on multi-head self-attention (Vaswani et al. 2017), where each hero has own embeddings and feed into the module to get the team representation for predicting match outcomes. Since we don't utilize any in-game feature except hero IDs, we remove the feature module of OptMatch. Besides, OptMatch assumes teams on two sides have the same size, therefore, we don't apply OptMatch on TFT dataset.

Model Variants

To examine the effectiveness of each component in NeuralAC, we conducted a series of ablation experiments.

- *no-coop*: A variant of NeuralAC that does not model the cooperation effect, i.e., remove f_1.
- *no-comp*: A variant of NeuralAC that does not model the competition effect, i.e., remove f_2.
- *no-att*: A variant of NeuralAC that all attention modules are removed, i.e., remove a_{ij}^{coop} and a_{ij}^{comp}.

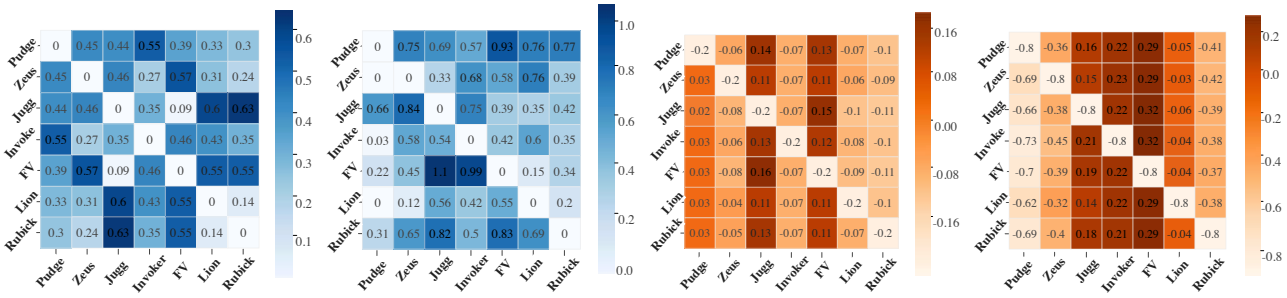
Experimental Setup

For NeuralAC model, the dimension of hidden layers is set to 50, and ReLu is used as activation function. We initialize the parameter with *Kaiming* initialization (He et al. 2015). Besides, Dropout (Srivastava et al. 2014) technique is also applied with the drop probability set to 0.2.

For every dataset, we randomly divided samples into 80% for training, 10% for validating, and 10% for testing. We choose Area Under ROC (AUC) (Bradley 1997) and Accuracy (Acc) as the evaluation metrics. For HOI and

Model	Dota2015		Dota2018		LoL		TFT	
	AUC	Acc	AUC	Acc	AUC	Acc	AUC	Acc
BT	0.6330	0.5955	0.6116	0.5784	0.6347	0.5969	0.7634	0.6935
LR	0.6330	0.5956	0.6116	0.5784	0.6347	0.5969	0.7634	0.6935
TrueSkill	0.6110	0.5789	0.5805	0.5577	0.6129	0.5811	0.7506	0.6832
LGB	0.6445	0.6035	0.6224	0.5929	0.6411	0.6028	0.8015*	0.7234*
HOI	0.6373	0.5989	0.6144	0.5821	0.6337	0.5965	0.7728	0.6989
OptMatch	0.6325	0.5961	0.6173	0.5851	0.6523	0.6101	-	-
NeuralAC	**0.6615**	**0.6156**	**0.6411**	**0.6012**	**0.6663**	**0.6209**	**0.8082**	**0.7279**
no-coop	0.6525	0.6086	0.6333	0.5951	0.6531	0.6110	0.7992	0.7215
no-comp	0.6444	0.6051	0.6203	0.5841	0.6546*	0.6115*	0.7740	0.7000
no-att	0.6606*	0.6150*	0.6396*	0.5991*	0.6480	0.6070	0.7780	0.7037

Table 2: Experimental results on match outcome prediction. (The second best methods are denoted with *)



(a) Cooperation scores. (b) Competition scores. (c) Cooperation attention values. (d) Competition attention values.

Figure 3: Note that the cooperation scores matrix is symmetry, the other three matrices are asymmetric since the impact of hero i on hero j is usually different from the impact of hero j on hero i. The diagonal is left blank because one cannot interact with himself. Take subfigure (b) for illustration, the value in i-th row and j-th column, denote the competition score when hero in i-th row competes against hero in j-th column.

NeuralAC, we set embedding size k to 20. As for OptMatch, the embedding size was searched in [20, 40, 80, 160, 320] to get the best performance on validating. We choose Adam (Kingma and Ba 2014) as the optimizer, with 0.001 of learning rate and 0.0001 of weight decay coefficient, for HOI, OptMatch and NeuralAC. Besides, the batch size is set to 256 for HOI, OptMatch and NeuralAC on all datasets.

LR, TrueSkill and LGB are implemented by open source packages sklearn, trueskill, LightGBM, respectively. HOI, NeuralAC and OptMatch are implemented by PyTorch package (Paszke et al. 2019). All experiments are implemented by Python and are trained on a Linux server with Intel Xeon E5-2650 CPUs and a TITAN Xp GPU.

Experimental Results

Table 2 shows the experimental results of all methods on the prediction task. First, NeuralAC outperforms all the other baselines on all datasets, indicating the effectiveness of our model. Second, by incorporating cooperation effects, HOI performs better than BT, LR and Trueskill on most datasets. This proves the existence of the cooperative effect in group comparison. Third, no-comp performs better than HOI, which indicates that the inner product may fail to model complex intra-team interactions. Fourth, no-comp outperforms OptMatch in 2 out of 3 datasets. One possible reason may be that despite OptMatch modeling higher-order interactions, it preserves little low-level information. Finally, compared with other variants, NeuralAC performs better, which suggests that incorporating competition effects, and attention mechanisms improves the accuracy of prediction.

Model Interpretability

To evaluate the interpretability of NeuralAC (i.e., whether the cooperation effects, competition effects and attention distribution are reasonable), we choose the most 7 popular heroes in Dota2018, then calculate their pair-wise cooperation scores, competition scores, and attention values separately. The corresponding results are shown in Figure 3.

Cooperation Score. Intuitively, if two individuals i and j perform better when they play together, they are more likely to get a higher cooperation score. Similarly, if i suppress j more when j is i's opponent, i is more likely to get a

	indivi. effects	coop. effects	comp. effects
T_A	0.5284	1.9886	3.5600 ($T_A \rightarrow T_B$)
T_B	0.5494	2.0271	2.2458 ($T_B \rightarrow T_A$)

Table 3: Effects of two teams.

T_A's competition scores			T_B's competition scores		
T_A	T_B	value	T_B	T_A	value
Spectre	Riki	1.5	Phoenix	Spectre	0.82
Spectre	Luna	1.1	Riki	Spectre	0.75
Pudge	Pugna	1.1	Pugna	Spectre	0.64
	...			...	
Spectre	Phoenix	0.26	Luna	Zeus	0
Zeus	Pugna	0.2	Skywrath	Pudge	0

Table 4: Competition scores of two teams.

higher competition score over j. As shown in Figure 3(a), Jugg or FV get a high score when they play with Lion or Rubick. One likely explanation may be that both Jugg and FV are melee Damage Per Second (DPS) heroes, which means they can deal huge physical damage to an enemy in a short time, but they have a small attack range; Lion and Rubick are ranged wizards with stun spells, which means they can help Jugg or FV get close to their enemy, hence improve attack efficiency. Two different types of heroes can complement each other. Therefore, Jugg cooperates well with Rubick or Lion. Furthermore, we can observe that the cooperation score between Jugg and FV is extremely low, which validates our assumption from another aspect because heroes of the same type cooperate poorly.

Competition Score. In the first columns of Figure 3(b), some values are relatively low, which suggests that Pudge almost immune from Zeus, Invoker, and Lion. Because Zeus, Invoker, and Lion rely heavily on magic spells while Pudge has a high magic resistance.

Attention Distribution. As shown in Figure 3(c) and Figure 3(d), FV and Jugg get relatively high attention values in two figures, because they are key characters in the game. One strange finding is Invoker get low scores in Figure 3(c), but high scores in Figure 3(d). A possible reason for this abnormality is Invoker has 10 unique spells, which make him one of the most powerful heroes in dota2. Unlike FV or Jugg, even without the assists of teammates, Invoker could still play an role in the battle. In the first column of Figure 3(d), Pudge get lowest attention values from enemy, since Pudge is one of the strongest heroes with high health, making him the last person his opponent wants to attack.

Through the above analysis, we can infer that NeuralAC indeed learns meaningful and reasonable relationships between heroes. It is worth to point out that our proposed model is capable to learn such complex relationships merely based on game outcomes, with no prior knowledge (e.g., attack range) of heroes[4] is provided to NeuralAC.

[4]To know more about heroes in Dota2, you can refer to the following websites: https://dota2.gamepedia.com/Heroes, https://www.dota2.com/heroes/?l=english

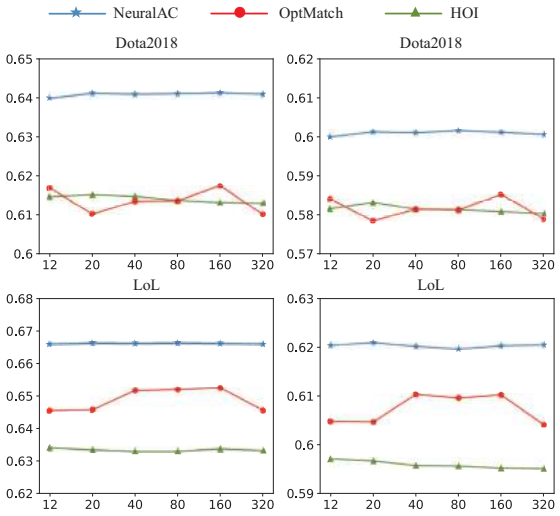


Figure 4: Test Acc and AUC w.r.t. embedding size k.

Case Study

Here we present a match record on dataset Dota2018, where T_A beats T_B in the end. As shown in Table 3, T_A and T_B get quite close individual effects and cooperation effects. However, the difference between their competition effects is huge. In Table 4, We detail the three highest competition scores and the two lowest competition scores for two teams. Overall, we can observe heroes in T_A have more edges when competing against heroes in T_B. If a method fail to model competition effects, then it may give a wrong prediction.

Hyperparameters Effects

Since HOI, OptMatch, and NeuralAC have embedding layers, we conduct a group of experiments on Dota2018 and LoL to explore the impact of the embedding size, where other parameters (e.g., batch size, learning rate) are fixed. As shown in Figure 4, it is obvious that NeuralAC consistently performs the best under all parameter settings. However, the model does not learn better when embedding size increases.

Conclusions

In this paper, we proposed NeuralAC for match outcomes prediction. By modeling both attentional cooperation effects and attentional competition effects with neural networks, NeuralAC outperforms the state-of-the-art methods. Extensive experimental results on four datasets showed the effectiveness of NeuralAC. Besides, we demonstrated that NeuralAC could be seen as the generalization of several previous models. Finally, NeuralAC provides meaningful and reasonable relationships between individuals, which can be further used in team formations (Wright and Vorobeychik 2015), hero recommendation, and user performance prediction (Huang et al. 2020; Wu et al. 2020; Wang et al. 2020).

Acknowledgements

We would like to thank RiotGame for providing convenient API. We also want to thank Linxia Gong for her advice on this research. This research was partially supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101), the National Natural Science Foundation of China (Grants No. 61922073 and 61672483), and the Foundation of State Key Laboratory of Cognitive Intelligence (Grant No. iED2020-M004).

References

Bar-Yam, Y. 2003. Complex systems and sports: complex systems insights to building effective teams. *Cambridge (MA): NECSI* .

Bengtsson, M.; and Kock, S. 1999. Cooperation and competition in relationships between competitors in business networks. *Journal of business & industrial marketing* .

Bradley, A. P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30(7): 1145–1159.

Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; and Chua, T.-S. 2017. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of CVPR*, 5659–5667.

Chen, S.; and Joachims, T. 2016. Modeling intransitivity in matchup and comparison data. In *Proceedings of WSDM*, 227–236.

Chen, Z.; Nguyen, T.-H. D.; Xu, Y.; Amato, C.; Cooper, S.; Sun, Y.; and El-Nasr, M. S. 2018. The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 200–208.

Dai, L.; Yin, Y.; Qin, C.; Xu, T.; He, X.; Chen, E.; and Xiong, H. 2020. Enterprise Cooperation and Competition Analysis with a Sign-Oriented Preference Network. In *Proceedings of SIGKDD*, 774–782.

Delalleau, O.; Contal, E.; Thibodeau-Laufer, E.; Ferrari, R. C.; Bengio, Y.; and Zhang, F. 2012. Beyond skill rating: Advanced matchmaking in ghost recon online. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3): 167–177.

DeLong, C.; Pathak, N.; Erickson, K.; Perrino, E.; Shim, K.; and Srivastava, J. 2011. TeamSkill: modeling team chemistry in online multi-player games. In *PAKDD*, 519–531. Springer.

Gong, L.; Feng, X.; Ye, D.; Li, H.; Wu, R.; Tao, J.; Fan, C.; and Cui, P. 2020. OptMatch: Optimized Matchmaking via Modeling the High-Order Interactions on the Arena. In *Proceedings of SIGKDD*, 2300–2310.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

He, X.; and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 355–364.

Herbrich, R.; Minka, T.; and Graepel, T. 2007. TrueSkill: a Bayesian skill rating system. In *NeurIPS*, 569–576.

Huang, T.-K.; Lin, C.-J.; and Weng, R. C. 2008. Ranking individuals by group comparisons. *Journal of Machine Learning Research* 9(Oct): 2187–2216.

Huang, Z.; Liu, Q.; Chen, Y.; Wu, L.; Xiao, K.; Chen, E.; Ma, H.; and Hu, G. 2020. Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students. *ACM Transactions on Information Systems (TOIS)* 38(2): 1–33.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *NeurIPS*, 3146–3154.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Li, Y.; Cheng, M.; Fujii, K.; Hsieh, F.; and Hsieh, C.-J. 2018a. Learning from group comparisons: exploiting higher order interactions. In *NeurIPS*, 4981–4990.

Li, Z.; Zhao, H.; Liu, Q.; Huang, Z.; Mei, T.; and Chen, E. 2018b. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of SIGKDD*, 1734–1743.

Liu, Q.; Huang, Z.; Huang, Z.; Liu, C.; Chen, E.; Su, Y.; and Hu, G. 2018. Finding similar exercises in online education systems. In *Proceedings of SIGKDD*, 1821–1830.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 6379–6390.

MacRae, S. A. 2018. Competition, cooperation, and an adversarial model of sport. *Journal of the Philosophy of Sport* 45(1): 53–67.

Minka, T.; Cleven, R.; and Zaykov, Y. 2018. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.* .

Ng, A. Y.; and Jordan, M. I. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NeurIPS*, 841–848.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 8026–8037.

Rendle, S. 2010. Factorization machines. In *ICDM*, 995–1000. IEEE.

Semenov, A.; Romov, P.; Korolev, S.; Yashkov, D.; and Neklyudov, K. 2016. Performance of machine learning algorithms in predicting game outcome from drafts in Dota 2. In *International Conference on Analysis of Images, Social Networks and Texts*, 26–37. Springer.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1): 1929–1958.

Tauer, J. M.; and Harackiewicz, J. M. 2004. The effects of cooperation and competition on intrinsic motivation and performance. *Journal of personality and social psychology* 86(6): 849.

Usmani, S.; Bernagozzi, M.; Huang, Y.; Morales, M.; Sarvestani, A. S.; and Srivastava, B. 2020. Clarity: Data-Driven Automatic Assessment of Product Competitiveness. In *Proceedings of AAAI*, volume 34, 13204–13211.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, F.; Liu, Q.; Chen, E.; Huang, Z.; Chen, Y.; Yin, Y.; Huang, Z.; and Wang, S. 2020. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of AAAI*, volume 34, 6153–6161.

Wray, K. H.; Kumar, A.; and Zilberstein, S. 2018. Integrated Cooperation and Competition in Multi-Agent Decision-Making. In *Proceedings of AAAI*, 4751–4758.

Wright, M.; and Vorobeychik, Y. 2015. Mechanism design for team formation. In *Proceedings of AAAI*, 1050–1056.

Wu, R.; Deng, H.; Tao, J.; Fan, C.; Liu, Q.; and Chen, L. 2020. Deep Behavior Tracing with Multi-level Temporality Preserved Embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2813–2820.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T.-S. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* .

Zhang, K.; Zhang, H.; Liu, Q.; Zhao, H.; Zhu, H.; and Chen, E. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of AAAI*, volume 33, 5773–5780.

Learning to Identify Top Elo Ratings: A Dueling Bandits Approach

Xue Yan[1,2], **Yali Du** * [3], **Binxin Ru** [4], **Jun Wang** [5], **Haifeng Zhang**[1,2], **Xu Chen**[6]

[1] Institute of Automation, Chinese Academy of Sciences, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[3] Department of Informatics, King's College London, UK
[4] Machine Learning Research Group, University of Oxford, UK
[5] Department of Computer Science, University College London, UK
[6] Gaoling School of Artificial Intelligence, Renmin University of China, China
yanxue2021@ia.ac.cn, yali.du@kcl.ac.uk, robin@robots.ox.ac.uk,
jun.wang@cs.ucl.ac.uk, haifeng.zhang@ia.ac.cn, xu.chen@ruc.edu.cn

Abstract

The Elo rating system is widely adopted to evaluate the skills of (chess) game and sports players. Recently it has been also integrated into machine learning algorithms in evaluating the performance of computerised AI agents. However, an accurate estimation of the Elo rating (for the top players) often requires many rounds of competitions, which can be expensive to carry out. In this paper, to improve the sample efficiency of the Elo evaluation (for top players), we propose an efficient online match scheduling algorithm. Specifically, we identify and match the top players through a dueling bandits framework and tailor the bandit algorithm to the gradient-based update of Elo. We show that it reduces the per-step memory and time complexity to constant, compared to the traditional likelihood maximization approaches requiring $O(t)$ time. Our algorithm has a regret guarantee of $\tilde{O}(\sqrt{T})$, sublinear in the number of competition rounds and has been extended to the multidimensional Elo ratings for handling intransitive games. We empirically demonstrate that our method achieves superior convergence speed and time efficiency on a variety of gaming tasks.

Introduction

In this paper, we investigate the selection of best multi-agent strategies under the Elo rating systems. The evaluation of the competition outcome has received lots of attention, especially in view of the successful usage of reinforcement learning in StarCraft (Vinyals et al. 2019; Han et al. 2019; Du et al. 2019), Game of Go (Silver et al. 2017) and video games (Mnih et al. 2015). The Elo rating system (Elo 1978) is a predominant and valuable algorithm for evaluating and ranking agents. In the widely adopted Bradley-Terry model (Hunter et al. 2004) for Elo, each player is assigned a numerical rating which is updated with competition outcomes via online stochastic gradient descent. Further, for dealing with non-transitive relations between interacting agents such as the game of *Rock-Paper-Scissors*, Balduzzi et al. (2018) proposes multidimensional Elo (mElo), which decomposes a game into transitive and cyclic parts to handle intransitive skills and evaluates different strategies by computing Nash-averaging.

In practical settings when a competition is expensive to conduct, updating Elo rating in a sample efficient way is highly valuable. To achieve such sample efficiency, we need a way to select the most informative pairs for evaluation. Two popular sampling approaches are Round-robin (Rasmussen and Trick 2008) and Elimination tournament (Groh et al. 2012); The Round-robin (Rasmussen and Trick 2008) is widely used in sport scheduling to balance the total time, venue usage and fairness of tournaments. It would arrange each team to play against all the others in as few as possible days while satisfying some constraints such as each team not playing twice in the same day to promote game fairness. By contrast, the Elimination tournament (Groh et al. 2012) only allows the winners at each round to proceed to the next round, so the stronger team will have the chance to play more times. A recent approach, RG-UCB, Rowland et al. (2019) introduces an adaptive sampling scheme to estimate the accurate ranking among all agents. RG-UCB considers sampling of agent match-ups as a collection of pure exploration bandit problems (Bubeck, Munos, and Stoltz 2011) and requires enough pairwise comparison for estimating each pair of strategies.

However, these tournament matching/sampling methods suffer from two major limitations which prohibit their wide usage in the modern large scale evaluations. Firstly, both the Round-robin and the Elimination tournament organise competitions following a pre-designed schedule, and the Elimination tournament scheduling may need some prior knowledge on the players' skill. Also each pair of players only compete once in both schemes so the results can be highly noisy. Secondly, the main idea behind the matching schemes of the RG-UCB and the Round-robin is random sampling, which fails to pay more attention to more promising players and/or pairs with higher uncertainty in competition outcome. Thus, they are less sample efficient in identifying the best players.

*Corresponding to Yali Du ⟨yali.du@kcl.ac.uk⟩.

In this work, we propose two sampling algorithms, named MaxIn-Elo and MaxIn-mElo, for the update of Elo and mElo rating systems respectively. Specifically, we maintain a candidate set with promising players using UCB-based (Upper Confidence Bound) dueling bandits and then select the pair with the highest uncertainty in competition outcome at each round. On the one hand, our algorithms are adapted to the gradient-based update of Elo rating systems, thus more memory and time efficient compared to a prior work, MaxInP (Saha and Gopalan 2020), which relies on maximum likelihood estimation (MLE). One the other hand, we extend our method to update mElo (multidimensional Elo) ratings to handle intransitive games, while Saha and Gopalan (2020) is based on a generalized linear model and can only fit to the transitive games. To the best of our knowledge, this is the first work that enables online gradient-based update for dueling bandits, and a theoretical guarantee on the cumulative regret is provided.Also compared to a previous dueling bandit method descending through randomly sampled gradients at each time step (Yue and Joachims 2009), our method, by selecting the pairs with higher information gains from a set of top candidates, is more sample efficient and are guaranteed to converge at $\tilde{O}(\sqrt{T})$[1].

In summary, our contributions are three-fold: Firstly, we are the first to propose two online active sampling algorithms MaxIn-Elo and MaxIn-mElo that select maximum informative pairs with dueling bandits to update Elo and mElo ratings. Secondly, we give the regret analysis of our proposed MaxIn-Elo and show that the regret converge at $\tilde{O}(\sqrt{T})$. Thirdly, we demonstrate empirically on synthetic and real-world games that our algorithms achieve significantly lower cumulative regret than all baselines. Notably, our methods outperform MaxInP which uses maximum likelihood for more accurate estimation while with lower time and memory complexity.

Related Work

Multi-agent evaluation has attracted wide attention in ranking of players (Silver et al. 2017; Lai 2015; Arneson, Hayward, and Henderson 2010; Gruslys et al. 2018) and in selecting stronger strategies in meta games (Muller et al. 2020; Czarnecki et al. 2020). There are many methods used for multi-agent evaluation problem. The Elo rating system is widely used for two-player games such as chess and tennis. It increases (decreases) player's rating according to player wins (loss) a competition, and updates ratings by online stochastic gradient descent (SGD), which is computationally efficient and simple to implement. While the Elo rating system cannot handle intransitive games such as rock-paper-scissors, multidimensional Elo (Melo) (Balduzzi et al. 2018) was introduced. It decomposes the win-loss matrix of an intransitive game into the transitive component and cyclic component baked in Hodge decomposition theory (Jiang et al. 2011). α-rank (Omidshafiei et al. 2019) is another popular counterpart in tackling intransitive games; recent attempts improve its sample efficiency based on noisy comparisons (Du et al. 2021; Rowland et al. 2019; Omidshafiei et al. 2019) and scalability by stochastic optimization (Yang et al. 2020). Despite various evaluation algorithms discussed, how to sample agent pairs at each round is of high value to realize these algorithms in large-scale evaluation tasks.

We consider dueling bandits for online match scheduling in the evaluation of players. The concept of dueling bandits was firstly proposed in (Yue and Joachims 2009). Compared to traditional bandits algorithms which pull one arm at each round and receive the reward of this arm directly, dueling bandits pull arm-pair at each round and only get a binary comparison result. DBGD (Yue and Joachims 2009) models a convex optimization problem as the dueling bandits problem which aims to find the best point in a convex space, and DBGD uses a random gradient as the direction of exploration for selecting the next arm-pair. Yue et al. (2012) formulates the best player identification as a dueling bandits problem with noisy comparison results and an underlying winning probability matrix, and proposes two algorithms as well as their corresponding regret bounds. These algorithms identify best arms based on the observed binary feedback however do not learn the player's skills (ratings), which is helpful in predicting future competition outcomes. Szörényi et al. (2015) regards the ranking of M alternatives (e.g. human players or agents) as a dueling bandits problem. They introduce the confidence interval of rating or winning probability into the dueling bandits problem, and design algorithms to identify the close-to-optimal item or to obtain the close-to-optimal whole ranking respectively. Saha, Koren, and Mansour (2021) studies the adversarial setting, in which the winning probability is non-stationary because players' skill may change over time. And they measure arms' abilities by estimating Borda score, however Borda score does not possess predictive power of future competition results and the algorithm of (Saha, Koren, and Mansour 2021) estimates Borda score only by simply calculating the frequency of wins. Heckel et al. (2019) gives an active ranking algorithm that can solve the top-k player identification problem and find the entire sequential ranking among all players.

While existing algorithms could rely on Borda score to update the rankings or design specific sort algorithms to obtain the ranking of items, they are not suitable for the Elo rating systems that adopt stochastic gradient descent to update ratings. Eearlier attempt (Ding, Hsieh, and Sharpnack 2021) proposes SGD-TS for contextual bandit problem, which learns parameters in the generalized linear model through online SGD instead and employs Thompson Sampling (TS) (Thompson 1933; Agrawal and Goyal 2012, 2013) to encourage exploration in arm-pulling. Compared to UCB-GLM (Li, Lu, and Zhou 2017) that adopt maximum likelihood estimators, SGD-TS achieves a similar theoretical cumulative regret bound, but lower time and memory complexity.However, no prior work has studied the SGD update in dueling bandits setting.

In this work, we will tame dueling bandits for the online match scheduling in the Elo rating system that adopts SGD. Saha and Gopalan (2020) proposes the algorithm that selects Maximum-Informative-Pair (MaxInP) for K-armed contex-

[1] $\tilde{O}$ ignores poly-logarithmic factors

tual dueling bandits. This algorithm utilizes the MLE to estimate parameter $\hat{\theta}$ and uses UCB (Auer, Cesa-Bianchi, and Fischer 2002) estimator to narrow down the set of candidate pairs from which the pair of arms with the maximum uncertainty is selected at each round. Our algorithms adopt a similar design as MaxInP Saha and Gopalan (2020) in calculating the uncertainty of a pair. However, we use an online batch SGD instead of MLE to update Elo rating, which is more time and memory efficient.

Methodology

Background

Suppose there are n players, the Elo rating system (Elo 1978) assigns a rating $r_x, x \in [n]$ to each player representing its skill. Let r^* denote the true ratings of n players. Our aim is to identify the best player among all n players:

$$x^* = \arg\max_{x \in [n]} r_x^* \tag{1}$$

Denote P as the true winning probability matrix, p_{xy} as the underlying groundtruth probability of x beating y. Based on the Bradley-Terry model (Hunter et al. 2004), the predicted probablility of player x winning y is

$$\hat{p}_{xy} = \sigma(r_x - r_y). \tag{2}$$

$\sigma(x)$ is a sigmoid function with $\sigma(x) = \frac{1}{1+e^{-x}}$. Elo ratings are updated by maximizing the likelihood of win-loss predictions which corresponds to minimizing the loss:

$$\ell_{\text{Elo}}\ (p_{xy}, \hat{p}_{xy}) = -p_{xy} \log \hat{p}_{xy} - (1 - p_{xy}) \log (1 - \hat{p}_{xy}). \tag{3}$$

At time t player x compete with player y with outcome o_{xy}^t: $o_{xy}^t = 1$ if x wins and $o_{xy}^t = 0$ otherwise. We can use o_{xy}^t to compute the gradient of Eq. (3) and update Elo by gradient descent:

$$r_x^{t+1} \leftarrow r_x^t - \eta \cdot \nabla_{r_x} \ell_{\text{Elo}}\ \left(o_{xy}^t, \hat{p}_{xy}^t\right) = r_x^t + \eta \cdot \left(o_{xy}^t - \hat{p}_{xy}^t\right). \tag{4}$$

Let T denote the total number of rounds, at each round $t \in [T]$, we adopt a system that will pull a pair of players $(x_t, y_t) \in [n] \times [n]$ and get comparison result $o_t(x_t, y_t) \sim \text{Bern}(p_{xy})$. The cumulative regret of T rounds is defined as

$$R(T) = \sum_{t=1}^{T} [r_{x^*}^* - \frac{1}{2}(r_{x_t}^* + r_{y_t}^*)]. \tag{5}$$

The definition is consistent with (Saha and Gopalan 2020). It measures the reward difference between the best arm and the two selected arms at each round.

MaxIn-Elo Algorithm

We use the notations below in the followed presentations.

- n: the number of players.
- τ: the batch size.
- r: a vector of n players' ratings.
- r^*: the true ratings of n players.
- $\hat{r}_t$: estimator by MLE with t round comparisons.
- $\tilde{r}_j$: the SGD estimator at batch j.

Algorithm 1: MaxIn-Elo: Dueling bandits with online SGD for top player identification.

Input: batch size τ, maximum number of rounds T, N players' strategies, parameters α, γ.
Output: output r

```
Randomly choose a pair to compare and record as x_t, y_t, o_t for t ∈ [τ]
V_{τ+1} = Σ_{t=1}^{τ} (e_{x_t} − e_{y_t})(e_{x_t} − e_{y_t})^T
Calculate the maximum-likelihood estimator r̂_τ by solving
    ∇_r Σ_{t=1}^{τ} ℓ_Elo (o_t, p̂(x_t, y_t)) = 0
Maintain convex set C = {r : ‖r − r̂_τ‖ ≤ 2}
for t = τ+1, τ+2, ..., T do
    if t%τ = 1 then
        j ← ⌊(t−1)/τ⌋ and η_j = 1/(αj)
        Calculate gradient ∇_r l_{j,τ}(r̃_{j−1}) through Eq. (4)
        Update ratings r̃_j through Eq. (8)
        Compute r̄ = (1/j) Σ_{=1}^{j} r̃
    end if
    Define a candidate optimal set S = {x | r̄_x − r̄_y +
        γ ‖e_x − e_y‖_{V_t^{-1}} > 0, ∀y ∈ [n]/{x}}
    Select a pair as:
        (x_t, y_t) = arg max_{(x,y)∈S} ‖e_x − e_y‖_{V_t^{-1}}.
    Let players (x_t, y_t) compete and observe o_t(x_t, y_t)
    Compute V_{t+1} = V_t + (e_{x_t} − e_{y_t})(e_{x_t} − e_{y_t})^T
end for
```

- $\bar{r} = \sum_{=1}^{j} \tilde{r}$: the average of previous SGD iterations.
- $\|x\| = \sqrt{x^\top x}$: the standard ℓ_2 norm.
- e_i: the i-th unit base vector, i.e., the i-dimension equals 1 and all other components equal 0.
- V_t: the history matrix recording previous t pulling information defined by $V_t = \sum_{i=1}^{t-1} (e_{x_i} - e_{y_i})(e_{x_i} - e_{y_i})^\top$.
- $\|x\|_V$: a special ℓ_2-norm associated with matrix V defined by $\|x\|_V = \sqrt{x^\top V x}$.
- $\mathcal{B}$ a neighborhood of r^* with $\mathcal{B} = \{r | \ \|r - r^*\| \leq 3\}$.
- $\mathcal{C}$: a neighborhood of $\hat{r}_\tau$ with $\mathcal{C} = \{r | \ \|r - \hat{r}_\tau\| \leq 2\}$.
- $\prod_{\mathcal{C}}(.)$: the projection operation defined by:

$$\prod_{\mathcal{C}}(r) = \hat{r}_\tau + \frac{2 * (r - \hat{r}_\tau)}{\min\{2, \|r - \hat{r}_\tau\|\}} \tag{6}$$

Algorithm Overview The main idea of our MaxIn-Elo algorithm is to maintain a candidate set of promising items via UCB and select the most informative pairs out of the set to evaluate at each round. Firstly, the ratings are initialized by maximizing likelihood of Eq. (3) on a batch of randomly sampled pairs with batch size τ. The solution is denoted as $\hat{r}_\tau$ and $\tilde{r}_0 = \hat{r}$. Then starting from round $t = \tau + 1$, we update $\tilde{r}_j$ every τ rounds by solving the following objective function

$$l_{j,\tau}(r) = \sum_{t=(j-1)\tau+1}^{j\tau} \ell_{\text{Elo}}\ (o_t, \hat{p}(x_t, y_t)). \tag{7}$$

The stochastic gradient update of $\tilde{r}_j$ reads

$$\tilde{r}_j \leftarrow \prod_{\mathcal{C}} \left(\tilde{r}_{j-1} - \eta_j \nabla_r l_{j,\tau}(\tilde{r}_{j-1})\right). \tag{8}$$

First, the strong convexity of the objective function is required for fast convergence, and if we select a suitable τ through Eq. (14), the aggregated objective function $l_{j,\tau}(r)$ is a α-strong convex function when $r \in \mathcal{B}$. Second, to ensure $\tilde{r}_j \in \mathcal{B}$, $\tilde{r}_j$ is projected into the convex set $\mathcal{C}$ (also discussed in the proof of Lemma 2).

For each update, a batch of pairs are selected that lead to maximal information gain. The UCB score of a pair is defined by:

$$h(x_t, y_t) = \bar{r}_{x_t} - \bar{r}_{y_t} + \gamma \left\| e_{x_t} - e_{y_t} \right\|_{V_t^{-1}}, \tag{9}$$

with the balance parameter γ. The specific V_t^{-1} norm $\gamma \left\| e_{x_t} - e_{y_t} \right\|_{V_t^{-1}}$ measures the uncertainty between two arms. The UCB estimator balances the exploitation and exploration through combining ratings estimation $\bar{r}$ and the uncertainty term.

At each round t, we obtain a set of optimal player candidates $\mathcal{S}$ with positive UCB scores:

$$\mathcal{S} = \{x | h(x, y) > 0, \ \forall\, y \in [n]/\{x\}\}. \tag{10}$$

From the candidate set $\mathcal{S}$, we then pull a pair of arms with highest uncertainty by

$$(x_t, y_t) = \arg\max_{(x,y)\, \in \mathcal{S}\times\mathcal{S}} \left\| e_x - e_y \right\|_{V_t^{-1}} \tag{11}$$

to induce sufficient exploration. A detailed algorithm of our MaxIn-Elo is shown in Algorithm 1.

Compared to MaxInP which uses MLE at each iteration, MaxIn-Elo uses SGD to update the Elo rating r as traditional Elo does. Thus, our method is more efficient in both computation and time, and simple to implement. Compared to RG-UCB which randomly selects a pair to evaluate, our MaxIn-Elo selects the maximum informative pair and trade-off exploration and exploitation. The online SGD update rating according to a batch of comparisons to ensuring the α-strong convexity of the objective function $l_{j,\tau}$, thus the selection of the batch size τ is important for balancing the α-strong convexity the and the computation complexity of mini-batch update. To our best knowledge, this is the first algorithm that allows stochastic gradient descent update in dueling bandits settings. See Table1 for a comparison on time and memory.

MaxIn-mElo Algorithm

To enable the rating system to handle the intransitive skills, we extend the online sampling algorithm to multidimensional Elo ratings (mElo) (Balduzzi et al. 2018). Baking in the Hodge decomposition theory (Jiang et al. 2011), mElo proposed to decompose the antisymmetric logits matrix of win-loss probabilities into a transitive component, i.e. gradient flow of rating vector, and a cyclic component to capture the intransitive relations. By learning a $2k$-dimensional vector c_x and a rating r_x per player, the win-loss prediction for mElo_{2k} is defined as:

$$\hat{p}_{xy} = \sigma\left(r_x - r_y + c_x^\top \cdot \Omega_{2k\times 2k} \cdot c_y\right). \tag{12}$$

Algorithms	Regret	Time Complexity	Memory
DBGD	$O(T^{2/3})$	$O(T)$	$O(n)$
RG-UCB	No	$O(T)$	$O(n)$
Random	No	$O(T)$	$O(n)$
MaxInP	$\tilde{O}(\sqrt{T})$	$O(nT^2 + n^2T)$	$O(nT)$
MaxIn-Elo	$\tilde{O}(\sqrt{T})$	$O(n^2T)$	$O(n^2)$

Table 1: Comparison of regret, time complexity and memory with other algorithms. Our MaxIn-Elo and the MaxInP achieve the lowest regret bound $\tilde{O}(\sqrt{T})$, but our MaxIn-Elo has lower time and memory complexity than the MaxInP.

where $\Omega_{2k\times 2k} = \sum_{i=1}^{k} (e_{2i-1}e_{2i}^\top - e_{2i}e_{2i-1}^\top)$.

The UCB estimate of a pair (x_t, y_t) for mElo then becomes:

$$h(x_t, y_t) = \bar{r}_{x_t} - \bar{r}_{y_t} + \bar{c}_x^T \Omega \bar{c}_y + \gamma \left\| e_{x_t} - e_{y_t} \right\|_{V_t^{-1}}. \tag{13}$$

Notice that compared to Elo ratings with $k = 0$ (Eq. (2)), mElo ratings assign a feature vector per player to approximated intransitive interactions. We present the details for the mElo ratings and Algorithm 2 for MaxIn-mElo in Appendix.

Regret Analysis

We give the cumulative regret bound of MaxIn-Elo, as for as we know, this is the first work that combines the online gradient update with dueling bandits and gives the cumulative regret of dueling bandits while being updated with SGD. We make a mild assumption on the link function σ.

Assumption 1. *Define* $c_\eta = \inf_{\{\|r - r^*\| \le \eta\}} \sigma'(r_x - r_y)$, *where* $(x, y) \in [n] \times [n]$, *and we assume* $c_3 > 0$.

This assumption is similar to that in (Ding, Hsieh, and Sharpnack 2021). Our main results rely on the following concentration events and the proofs of which are deferred to Appendix.

Lemma 1. *Suppose we sample a sequence of arm pairs* $\{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\}$ *through Algorithm 1 up to round t, and assume the selected batch size* τ *satisfy that* $\lambda_{\min}(V_{\tau+1}) \ge 1$, *where* $\lambda_{\min}(V_{\tau+1})$ *means the minimum eigenvalue of the matrix* $(V_{\tau+1})$, *Then* $\forall t > 0$,

$$\sum_{i=\tau+1}^{\tau+t} \left\| (e_{x_i} - e_{y_i}) \right\|_{V_i^{-1}} < \sqrt{2nt \log\left(\frac{2\tau + t}{n}\right)}.$$

Lemma 1 gives the bound of the sum of selected pair's uncertainty from round $\tau + 1$ to t. And this lemma will be adopted to derive the cumulative regret bound. In the following Lemma 2, we show that when the batch size τ is chosen as Eq. (14), we have the concentration property of the averaged SGD estimator $\bar{r}$.

Lemma 2. *Assume that there exists a positive constant* λ_f *such that* $\lambda_{\min}\left(\mathbb{E}[(e_{x_t} - e_{y_t})(e_{x_t} - e_{y_t})^T]\right) \ge \lambda_f$ *holds at each round* $t > \tau$, *where* (x_t, y_t) *is sampled through Algo-*

rithm 1. Let the batch size τ satisfies

$$\tau_1 = 2\left(\frac{C_1\sqrt{n} + C_2\sqrt{2\log T}}{\lambda_{\min}(B)}\right)^2 + \frac{16(n + 2\log T)}{c_1^2\lambda_{\min}(B)},$$
$$\tau_2 = 2\left(\frac{C_1\sqrt{n} + C_2\sqrt{2\log T}}{\lambda_f}\right)^2 + \frac{4\alpha}{c_3\lambda_f},$$
$$\tau = \lceil \max\{\tau_1, \tau_2\} \rceil, \tag{14}$$

where $B = \mathbb{E}_{(x,y)\overset{iid}{\sim}[n]\times[n]}\left[(e_x - e_y)(e_x - e_y)^T\right]$. Define $g_1(t)$ and $g_2(j)$,

$$g_1(t) = \frac{1}{2c_1}\sqrt{\frac{n}{2}\log\left(1 + \frac{2t}{n}\right) + 2\log T}, \tag{15}$$

$$g_2(j) = \frac{\tau}{\alpha}\sqrt{1 + \log j}. \tag{16}$$

For a constant $\alpha \geq c_3$, there exists two positive constants C_1, C_2 such that if the batch size τ is chosen as Eq. (14)*, then we have that at each round $t > \tau$ corresponding to batch $j = \lfloor \frac{t-1}{\tau} \rfloor$, event $E_1(t)$ holds with probability at least $1 - \frac{5}{T^2}$, where $E_1(t) = \{\forall(x, y) : \left|(e_x - e_y)^T(\bar{r}_j - r^*)\right| \leq g_1(j\tau)\|e_x - e_y\|_{V_{j\tau+1}^{-1}} + g_2(j)\frac{\sqrt{2}}{\sqrt{j}}\}$.*

The following Lemma 3 shows how to select a suitable balanced parameter γ of UCB score that ensures the best player is always in the candidate set.

Lemma 3. *Define the constant $C = \sqrt{2nT\log\left(\frac{T+\tau}{n}\right)}$. At each round $t > \tau$, let UCB balanced parameter $\gamma = 2g_1(t)$ and assume $\Delta > g_1(T)C$, if α satisfies that $\alpha \geq \frac{\sqrt{2}\tau\sqrt{1+\log j}}{(\Delta - g_1(T)C)\sqrt{j}}$, then we have $x^* \in \mathcal{S}$ holds with probability at least $1 - \frac{5}{T^2}$, where $j = \lfloor \frac{t-1}{\tau} \rfloor$, Δ is the difference between ratings of optimal player x^* and sub-optimal player x'. Recall $x^* = \arg\max_{x\in[n]} r_x^*$, and define $x' = \arg\max_{x\in[n]/x^*} r_x^*$, $\Delta = r_{x^*}^* - r_{x'}^*$.*

Lemma 3 shows that if we properly select UCB balanced parameter γ and parameter α which describes objective function $l_{j,\tau}$ as a α-strongly convex, then it is promised that the best player x^* is in candidates set $\mathcal{S}$ with high probability. This property is helpful for the top-1 identification because the candidate set $\mathcal{S}$ will become tighter with the time, and x^* always in $\mathcal{S}$, thus candidate set $\mathcal{S}$ only contains x^* eventually. Together we are ready to present our main results in Theorem 1.

Theorem 1. *We run our Algorithm 1 to get a sequence of arm-pair, and let the learning rate parameter $\alpha \geq \max\{c_3, \frac{\sqrt{2}\tau\sqrt{1+\log j}}{(\Delta - g_1(T)C)\sqrt{j}}\}$ with assumption that $\Delta > g_1(T)C$, the balanced parameter $\gamma = 2g_1(t)$, there exists two positive parameter C_1, C_2 such that if the batch size τ is chosen as Eq.* (14)*, then we have the cumulative regret satisfies that:*

$$R(T) \leq \tau * \Delta_{max} + (2 + \tau)g_1(T)\sqrt{2nT\log(\frac{2\tau + T}{n})} + 4g_2(J)\sqrt{\tau T},$$

with probability at least $1 - \frac{10}{T}$, where $J = \lfloor \frac{T}{\tau} \rfloor$, $\Delta_{max} = \max_i r_i^* - \min_i r_i^*$, $g_1(T), g_2(J)$ is defined in Eq. (15) *and C is a constant defined as $C = \sqrt{2nT\log\left(\frac{T+\tau}{n}\right)}$.*

Note that $\tau \sim O(\max\{n, \log T\})$ (Eq. (14)), $g_1(T) \sim O(\sqrt{n\log T})$, $g_2(J) \sim O(\sqrt{\log T})$. Combining the above analysis, we have $R(T) \sim O(n\log T\sqrt{T})$ (or $\tilde{O}(\sqrt{T})$. This regret upper bound is equivalent to that in (Saha and Gopalan 2020) which employs MLE estimators. However, our algorithm improves the efficiency in terms of memory and time. The memory cost is constant with respect to T while MaxInP's memory cost is linear in the time horizon T. The time complexity of our MaxIn-Elo is $O(n^2T)$, while MaxInP's time complexity is $O(nT^2 + n^2T)$. See Table 1 for a detailed comparison. Detailed proofs are referred to Appendix.

Experiments

We consider the following two batteries of experiments to evaluate the performance of our algorithms in the scenarios of transitive and intransitive real world meta-games. Ablation studies of parameter γ, dimension of mElo and the batch size τ can be found in Appendix.

Baselines

Random: The pairwise matching scheme of the classical Round-robin (Rasmussen and Trick 2008) tournament is based on random sampling. We construct a simple baseline that randomly select a pair from all $n * (n-1)/2$ pairs with replacement. After sampling a pair, we use the Elo/mElo model to update the ratings.

RG-UCB (Rowland et al. 2019): This algorithm adopts a pure exploration sampling scheme, which uniformly samples a pair from the set containing pairs that need to be estimated. And the stopping condition $C(\delta)$ controls the total number of comparisons of each pair, where δ is a hyper parameter deciding the confidence level of estimated competitive results.

DBGD (Yue and Joachims 2009): This dueling bandits algorithm is popular in ranking tasks when only pair-wise binary feedback is available. It maintains one winning arm at each round, and randomly synthesizes a gradient to obtain the opponent arm in the contextual bandit setting. In our feature free setting, this is equivalent to randomly selecting a player as the opponent.

α-IG (Rashid, Zhang, and Ciosek 2021): This is an active sampling algorithm used for estimating the α-rank (Omidshafiei et al. 2019). This algorithm selects a pair with largest information gain at each round. In the transitive case, the top player has an α-rank score equal to 1. Due to the high computation cost at each round (computing α-rank for 80000 times in a 4×4 game), we only compare with it in a 4×4 transitive game: the '2 Good, 2Bad' game given by α-IG.

MaxInP (Saha and Gopalan 2020): This algorithm is for the generalized linear contextual dueling bandits problem, in which arms are represented as feature vectors. At each round t, it uses MLE to estimate model parameters θ relying on all historical comparisons. This algorithm calculates a candidate set containing advanced arms and pulls an arm-pair with the largest uncertainty. In order to fit their model, each player is described as a one-hot vector, and the estimated parameters θ correspond to players' ratings in our setting.

MaxIn-Elo: Our first algorithm adopts dueling bandits to adaptively sample pairs for Elo rating update in Eq. (2). The aim of our MaxIn-Elo is to identify the advanced players gradually, and to minimize the cumulative regret described in Eq. (5) simultaneously.

MaxIn-mElo: Our second algorithm tames the intransitive scenarios. Different to MaxIn-Elo, there is an extra vector c to capture intransitive relationship in competition outcome prediction. The dimension of c is set to 8 in experiments. For the MaxIn-mElo algorithm, we hope to identify players with superior mElo ratings and to minimize cumulative regret on mElo ratings.

Experiments Setting

Real world games We do our experiments on twelve real-games released by Czarnecki et al. (2020), most of which are implemented on the OpenSpiel framework (Lanctot et al. 2019). The six games used for evaluating Elo are Triangular game, Transitive game, Elo game, and three noisy variants of Elo games. The first three are transitive games; the three variants of Elo game are Elo games with additive Gaussian noises. The six intransitive games used for evaluating mElo are Kuhn-poker, AlphaStar, tic_tac_toe, hex, Blotto and 5,3-Blotto game.

The intransitivity of games can be revealed by sink strongly connected components (SSCCs) (Omidshafiei et al. 2019), which is a set of strategies that cannot be defeated by external strategies and all internal strategies become a circle, such as Rock, Paper, Scissors. The statistics of these games is shown in Table 2 in Appendix.

Metrics Except the cumulative regret defined in Eq. (5), we introduce three other metrics for Reciprocal Rank (RR), Normalized Discounted Cumulative Gain (NDCG), and Hit Raio (HR). RR is used for the results on generating top-1 players. NDCG and HR report discrete performance for top-1 performance and are thus used in top-k results.

Reciprocal Rank (RR) (Donmez, Svore, and Burges 2009) give the reciprocal of predicted ranking of the best player x^*. Define $RR = 1/R(x^*)$, where $R(x)$ returns the ranking of player x relying on currently predicted ratings $\bar{r}$. Larger RR corresponds to better performance on the top-1 player identification.

Hit Ratio@K (He et al. 2015) is defined as the ratio of the predicted top-k that belong to the true top-k. Since hit ratio does not consider the positions of correct predictions, we also adopt NDCG (Donmez, Svore, and Burges 2009) which assigns higher importance to top ranks. Normalized Discounted Cumulative Gain (NDCG) is widely used in the evaluation of rankings and NDCG@k measures the importance of predicted top-k players. It is given by

$$\text{NDCG@}K = \frac{1}{N_K} \sum_{i=1}^{K} \frac{2^{l(d_i)} - 1}{\log(i+1)},$$

where N_k is a normalizer to ensure that the perfect ranking would result in NDCG@$K = 1$. d_i denote the index of predicted i-th player, and $l(x) \in \{0, 1\}$ is the relevance level about top-k identification, we set $l(x) = 1$ if player x in true top-k otherwise 0.

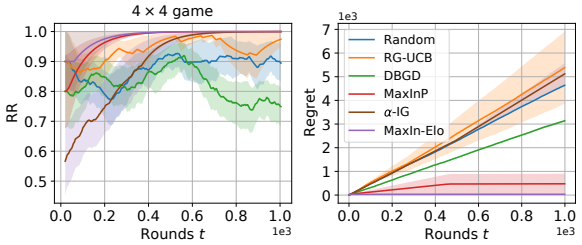


Figure 1: Results on 4×4 game (2 Good 2 Bad).

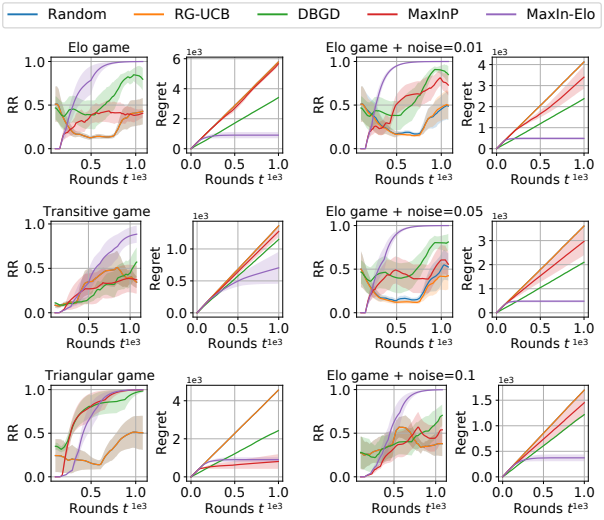


Figure 2: Results of Elo on transitive games.

Parameters setting For Random, DBGD, and RG-UCB baseline, we perform a grid search for the initial step size η in the range $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. For RG-UCB, stopping confidence $\delta = 0.2$. For MaxInP, we tune the UCB balanced parameter $\gamma \in \{0.2, 0.4, 0.6, ..., 2.0\}$. For MaxIn-Elo and MaxIn-mElo, we tune the initialized learning rate $\eta \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, and the learning rate at batch j is set as $\frac{\eta}{j}$. And the UCB balanced parameter $\gamma \in \{0.2, 0.4, 0.6, ..., 2.0\}$. The batch size τ of MaxInP, MaxIn-Elo and MaxIn-mElo is set to $0.7 * n$. When baselines ues mElo model to calculate ratings, we set the dimension of the extra vector c as 8. We use the parameters that report the best performance for α-IG. We repeat experiments 5 times with different random seeds and plot the averaged performance with standard deviations.

All experiments were run in a single x86_64 GNU/Linux machine with 256 AMD EPYC 7742 64-Core Processor and 2 A100 PCIe 40GB GPU. We use sklearn(0.24.2) to solve the MLE.

Results

Figure 1, 2, 3 show the results of top-1 identification on 13 games. To ensure a fair comparison between all baselines, we perform a grid search to select parameters with the best RR performance for each random seed. If the winning probability matrix can be fitted into the Elo model, then we cal-

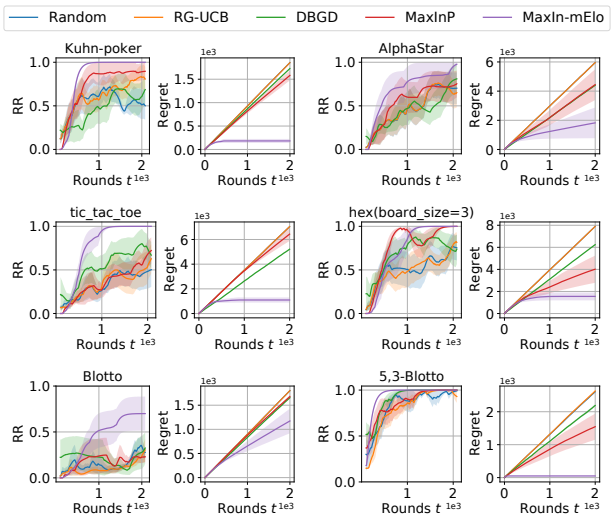


Figure 3: Results of mElo on intransitive games.

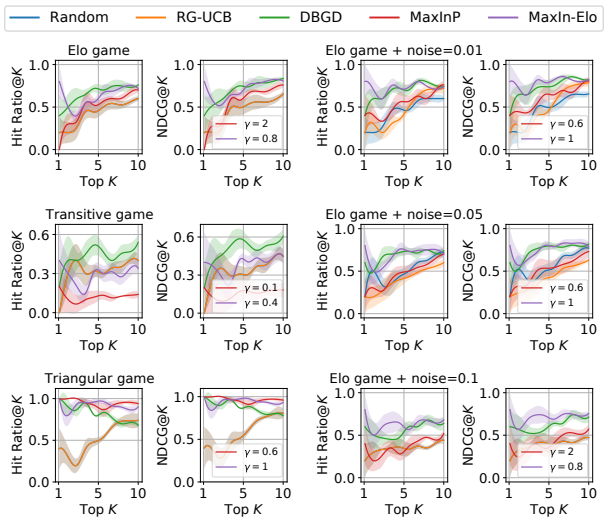


Figure 4: Results of Top-k player identification on transitive games. γ in red and purple indicates that reports best performance for MaxInP and MaxIn-Elo respectively.

culate the true ratings through Eq. (2), otherwise we use the mElo ratings as the true ratings through Eq. (12).

Evaluation of MaxIn-Elo Figure 1 shows the results of a 4×4 transitive game. MaxIn-Elo has the highest convergence rate on both RR and cumulative regret metrics, and MaxIn-Elo has the lowest cumulative regret close to 0. As shown in Figure 2, MaxIn-Elo significantly outperforms all other baselines on five games and achieves similar performance on Triangular game. Regarding the RR metric, MaxIn-Elo can converges to 1 on four games. Even on Transitive game and Elo game + noise=0.1, RR scores as up to 0.6 and 0.8 respectively, which indicates that the rank of the top player is no more than 2. Thus we think MaxIn-Elo has the ability to effectively identify the top player . On the Elo game, Elo game + noise=0.01, and Elo game + noise=0.05, the cumulative regret is closed to convergence at around 500 rounds. When the cumulative regret meets convergence, the candidate optimal set $\mathcal{S}$ only contains the top player, and no regret increasing.

Different from the other 5 stochastic games, Triangular game is a deterministic game with all winning probabilities are equal to 1 or 0, thus it is easy to evaluate. For DBGD baseline, it maintains the current best player and randomly selects an opponent, so it could find the best player more quickly, but has a large cumulative regret because of randomly selected opponents.

Evaluation of MaxIn-mElo Figure 3 shows the results of baselines on six real-world intransitive games. MaxInP is based on the Elo model for it is a special generalized linear model only with rating parameter r without cyclic vector parameter c, but all other baselines are based on the mElo model. As the Figure 3 shows, MaxIn-mElo has the lowest cumulative regret and the highest RR on all six games. With regard to the RR, MaxIn-mElo can be up to 1 on all games except for Blotto. One possible reason why MaxIn-mElo cannot be up to 1 on Blotto may be that its size of top SSCC is very large. The other reason is that we use the low-rank approximation of the probability matrix's rotation on the mElo model. Although we misidentified the top-1 player, we are still better than all other baselines.

Results of Top-k player identification Figure 4 gives the results of top-k predictions on transitive games. MaxIn-Elo and MaxInP both have a parameter γ used to balance exploration and exploitation, larger γ can lead to a larger candidate set then lead to better top-k performance. We keep other parameters fixed and run experiments with different $\gamma \in \{0.2, 0.4, 0.6, ..., 2.0\}$, and we report the performance of MaxInP and MaxIn-Elo under the best γ. Figure 4 shows that MaxIn-Elo has the best performance of the top-1 identification on all games, and it achieves the comparable performance of top-k identification on most games. Results of different γ can be found in Appendix.

Discussions

This work studied the problem of multi-agent evaluation with Elo ratings. We have adopted an online match scheduling framework to improve the sample efficiency of the Elo rating system and its extension mElo for the intransitive settings. Both empirical and theoretical results justify that our algorithms can achieve higher sample efficiency and lower regret on most of the tasks.

We consider two limitations of this work. Firstly, the match outcome prediction in our algorithm is based on only ratings without considering features that describe players. Future work may consider adding features into the match prediction. Secondly, our algorithm focuses more on identifying the best player without being tailored for identifying top-k players. Future work can consider active sampling that achieves better results on both top-1 and top-k cases.

Ethics Statement

This work proposes algorithms for online match scheduling that improve the efficiency in identifying top players in competitive games such as chess. While empirical studies in this work, which are based on AI agents, have demonstrated the superior gain of using our proposed methods, there is a caveat that our algorithms assume that the all players' skill levels remain unchanged throughout the repeated competition rounds. This assumption likely does not hold for human players whose playing strengths will be affected by energy consumption due to frequent matches. Therefore, extra caution needs to be taken when deploying our methods to schedule real-world competitions involving human players and an interesting research extension would be to model such performance strength changes explicitly in designing the match scheduling algorithms.

Acknowledgements

Co-author Haifeng Zhang is supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA27030401.

References

Agrawal, S.; and Goyal, N. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, 39–1.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning (ICML)*, 127–135.

Arneson, B.; Hayward, R. B.; and Henderson, P. 2010. Monte Carlo tree search in Hex. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4): 251–258.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2): 235–256.

Balduzzi, D.; Tuyls, K.; Perolat, J.; and Graepel, T. 2018. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3268–3279.

Bubeck, S.; Munos, R.; and Stoltz, G. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19): 1832–1852.

Czarnecki, W. M.; Gidel, G.; Tracey, B.; Tuyls, K.; Omidshafiei, S.; Balduzzi, D.; and Jaderberg, M. 2020. Real World Games Look Like Spinning Tops. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

Ding, Q.; Hsieh, C.-J.; and Sharpnack, J. 2021. An efficient algorithm for generalized linear bandit: Online stochastic gradient descent and thompson sampling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1585–1593.

Donmez, P.; Svore, K. M.; and Burges, C. J. 2009. On the local optimality of lambdarank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 460–467.

Du, Y.; Han, L.; Fang, M.; Dai, T.; Liu, J.; and Tao, D. 2019. LIIR: learning individual intrinsic reward in multi-agent reinforcement learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, 4403–4414.

Du, Y.; Yan, X.; Chen, X.; Wang, J.; and Zhang, H. 2021. Estimating α -Rank from A Few Entries with Low Rank Matrix Completion. In *International Conference on Machine Learning (ICML)*, 2870–2879. PMLR.

Elo, A. E. 1978. *The rating of chessplayers, past and present*. Arco Pub.

Groh, C.; Moldovanu, B.; Sela, A.; and Sunde, U. 2012. Optimal seedings in elimination tournaments. *Economic Theory*, 49(1): 59–80.

Gruslys, A.; Dabney, W.; Azar, M. G.; Piot, B.; Bellemare, M.; and Munos, R. 2018. The Reactor: A fast and sample-efficient Actor-Critic agent for Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

Han, L.; Sun, P.; Du, Y.; Xiong, J.; Wang, Q.; Sun, X.; Liu, H.; and Zhang, T. 2019. Grid-wise control for multi-agent reinforcement learning in video game ai. In *International Conference on Machine Learning (ICML)*, 2576–2585. PMLR.

He, X.; Chen, T.; Kan, M.-Y.; and Chen, X. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, 1661–1670.

Heckel, R.; Shah, N. B.; Ramchandran, K.; Wainwright, M. J.; et al. 2019. Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6): 3099–3126.

Hunter, D. R.; et al. 2004. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1): 384–406.

Jiang, X.; Lim, L.-H.; Yao, Y.; and Ye, Y. 2011. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1): 203–244.

Lai, M. 2015. Giraffe: Using deep reinforcement learning to play chess. *arXiv preprint arXiv:1509.01549*.

Lanctot, M.; Lockhart, E.; Lespiau, J.-B.; Zambaldi, V.; Upadhyay, S.; Pérolat, J.; Srinivasan, S.; Timbers, F.; Tuyls, K.; Omidshafiei, S.; et al. 2019. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning (ICML)*, 2071–2080.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; and Ostrovski, G. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529.

Muller, P.; Omidshafiei, S.; Rowland, M.; Tuyls, K.; Perolat, J.; Liu, S.; Hennes, D.; Marris, L.; Lanctot, M.; Hughes,

E.; et al. 2020. A Generalized Training Approach for Multiagent Learning. In *International Conference on Learning Representations (ICLR)*, 1–35.

Omidshafiei, S.; Papadimitriou, C.; Piliouras, G.; Tuyls, K.; Rowland, M.; Lespiau, J.-B.; Czarnecki, W. M.; Lanctot, M.; Perolat, J.; and Munos, R. 2019. α-rank: Multi-agent evaluation by evolution. *Scientific reports*, 9(1): 1–29.

Rashid, T.; Zhang, C.; and Ciosek, K. 2021. Estimating α-Rank by Maximizing Information Gain. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 5673–5681.

Rasmussen, R. V.; and Trick, M. A. 2008. Round robin scheduling–a survey. *European Journal of Operational Research*, 188(3): 617–636.

Rowland, M.; Omidshafiei, S.; Tuyls, K.; Perolat, J.; Valko, M.; Piliouras, G.; and Munos, R. 2019. Multiagent evaluation under incomplete information. In *Advances in Neural Information Processing Systems (NeurIPS)*, 12291–12303.

Saha, A.; and Gopalan, A. 2020. Regret Minimization in Stochastic Contextual Dueling Bandits. *arXiv preprint arXiv:2002.08583*.

Saha, A.; Koren, T.; and Mansour, Y. 2021. Adversarial dueling bandits. In *International Conference on Machine Learning (ICML)*, 9235–9244.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; and Bolton, A. 2017. Mastering the game of Go without human knowledge. *Nature*, 550(7676): 354.

Szörényi, B.; Busa-Fekete, R.; Paul, A.; and Hüllermeier, E. 2015. Online rank elicitation for plackett-luce: A dueling bandits approach. In *Proceedings of the 19th international Conference on Neural Information Processing systems (NeurIPS)*.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4): 285–294.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Yang, Y.; Tutunov, R.; Sakulwongtana, P.; and Ammar, H. B. 2020. α^α-Rank: Practically Scaling α-Rank through Stochastic Optimisation. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1575–1583.

Yue, Y.; Broder, J.; Kleinberg, R.; and Joachims, T. 2012. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5): 1538–1556.

Yue, Y.; and Joachims, T. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 1201–1208.

RESEARCH-ARTICLE

NBA Winner Prediction: A Hybrid Framework Incorporating Internal and External Factors

XI ZHENG, Xi'an Jiaotong University, Xi'an, Shaanxi, China

Published: 26 May 2022

BDE 2022: 2022 4th International Conference on Big Data Engineering
May 26 - 28, 2022
Beijing, China

https://doi.org/10.1145/3538950.3538960
ISBN: 9781450395632

NBA Winner Prediction: A Hybrid Framework Incorporating Internal and External Factors

Xi Zheng*
School of Management, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049
China2194420833@stu.xjtu.edu.cn

ABSTRACT

In recent years, extensive analysis has been applied to predicting NBA game results due to the popularity of basketball and massive financial transactions in NBA betting. The primary objective of this research is to construct a predictive model to precisely forecast the outcome of NBA basketball games in the latest 2020-21 and 2021-22 NBA regular season. We designed features which incorporates both external and internal factors such as teams' Elo rating, average team performance in recent games, home court advantage and tiredness due to back-to-back games. We built up three feature sets and performed feature selection using sequential feature selection (SFS) and recursive feature elimination (RFE) to verify their effectiveness. Results show that novel features such as level of tiredness and difference of Elo ratings between home and away team improves prediction accuracy. To make fair prediction for the latest NBA season, we utilize 10-fold cross validation to train and select models with decent mean accuracy and low standard deviation for final evaluation. It is found that our best random forest model performs fairly well in predicting games in the latest 2020-21 and 2021-22 season with an accuracy of 67.98%. The prediction results and the identification of key features that exert the most significant effects on the results can be helpful and meaningful to different stakeholders in this field, such as team coaches, players and NBA betters.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Learning paradigms**; • **Supervised learning**; • **Supervised learning by classification**;

KEYWORDS

NBA winner prediction, Sequential forward selection, Recursive feature elimination, Feature engineering, Feed forward neural network, Random Forest, Naïve Bayes

ACM Reference Format:
Xi Zheng. 2022. NBA Winner Prediction: A Hybrid Framework Incorporating Internal and External Factors. In *2022 4th International Conference on Big Data Engineering (BDE 2022), May 26–28, 2022, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3538950.3538960

*Corresponding author

BDE 2022, May 26–28, 2022, Beijing, China

ACM ISBN 978-1-4503-9563-2/22/05…$15.00
https://doi.org/10.1145/3538950.3538960

1 INTRODUCTION

The National Basketball Association (NBA) composed of 30 teams is a professional basketball league in the United States. The NBA regular season usually runs from November to April, which is 1230-game long, with each team playing 82 games, 41 at home and 41 away from home. From April to June, the top 8 teams that stand out from the regular season participate in the playoffs where the league champion is determined.

In recent years, extensive analysis has been applied to understand what contributes to a champion team and better predict future games. Cao (2012) constructed a predictive model based on Naïve Bayes Classifier. The classification accuracy of the test dataset was about 65.82% [1]. Thabtah et al (2019) discovered influential features set that affects the outcomes of NBA games based on several machine learning methods such as decision tree, artificial neural network and naïve bayes [2]. They found that the key features that help making better predictions are DRB(Defensive Rebounds), TPP(Three-Point Percentage), FT(Free Throw), and TRB(Total Rebounds), all of which subsequently increased the prediction accuracy by 2-4%. Based on players' statistics, Nguyen et al.(2021) utilized machine learning (ML) and deep learning (DL) to predict players' future performance and whether they will be selected in All-Star game [3]. Their study shows the performance of DL algorithms is not as good as ML on structured, relatively small-scale basketball datasets. Besides scholarly articles, professional sports forecasting companies also announced their accuracies on game results prediction across various NBA seasons: FiveThirtyEight correctly predicted the winner of 66.42% games during the 2017-18 NBA season [5], while NBA Miner correctly predicted the winner of 65.30% games during the 2015-16 seasons [6]. Generally, a good performance rate is seen between 65%-70%.

Most past researches utilized historical in-game statistics of teams, players or both for prediction. However, the results of NBA games are not only determined by intrinsic team competencies, but also an interplay of external factors, such as home court advantage, tiredness due to successive games and long trips between these games. Besides, few studies have researched the optimal time range of historical data that should be used for prediction, that is, how many past games should be taken into consideration when evaluating the team's recent performance for making accurate game result predictions. To fill up the void and make better predictions, in this article, we utilized not only historical box scores and in-game statistics such as ORB(Offensive Rebounds) and FT(Free Throw), but also the location of game and level of tiredness for prediction. Besides, a series of novel features that emphasize the difference of team statistics between home and away team are also created and

Table 1: Abbreviations for Basketball In-Game Statistics

Full Name	Abbreviation
Points	PTS
Total Rebounds	TRB
Offensive Rebounds	ORB
Defensive Rebounds	DRB
Assists	AST
Turnovers	TOV
Steals	STL
Blocks	BLK
All Free Throws	FT
Live Free Throws	LFT
Field Goals Attempts	FGA
Field Goals Made	FG
Field Goals Missed	FGM
2-Points Field Goals Attempts	2FGA
2-Points Field Goals Made	2FG
2-Points Field Goals Missed	2FGM
3-Points Field Goals Attempts	3FGA
3-Points Field Goals Made	3FG
3-Points Field Goals Missed	3FGM

used for prediction. Our results showed that adding these novel features into predictive models increased the prediction accuracy by 1.5%-2%. We further applied feature selection based on sequential forward selection (SFS) and recursive feature elimination (RFE) algorithm to identify the optimal feature subsets that achieve the highest prediction accuracy. Based on historical data from 2012-13 season to 2020-21 season, our optimal model gives a prediction accuracy of 67.98% in predicting the latest 2020-21 and 2021-22 NBA regular season.

This paper is outlined as follows: Section 1 presents the objectives, related articles, and the framework. Section 2 introduces data source and preprocessing process. Section 3,4 discuss feature engineering and the exploratory data analysis. Section 5,6 introduce the three feature sets and the feature selection methods. Section 7,8 present the selection of models as well as the final evaluation of top models. Section 9,10 draw conclusions and discuss possible future works.

2 DATA

2.1 Data Source

Due to the prevalence of sport data science and immense popularity of basketball and NBA games, there are various types of open-source NBA data that can be utilized for sports analytics and predictions, i.e. https://www.nba.com/, https://www.basketball-reference.com/ and https://www.espn.com/. The dataset used in this article is collected in December 2021 from Synergy Sports (https://synergysports.com/., which is a professional basketball database partners with NBA, WNBA, NCAA to create web-based, on-demand video-supported basketball analytics for the purposes scouting, development and entertainment. Synergy Sports also provides subscribers with videos of each game and an extensive range of detailed indicators including season summary, game situations, team performances, individual player behaviours, which offers valuable first-hand resources for advanced and deep investigation into the nature, laws and trends of basketball.

2.2 Data Preprocessing

Since the scraped data is stored by season, the first step of preprocessing is to combine datasets of different seasons into one dataset. The combined dataset comprises of 10364 rows and 209 columns of data. Besides, since each game comprises of 4 sections (12 minutes each) and the in-game statistics in the scraped datasets are measured on a per section basis which are too detailed for analytics, we add up the statistics of each section to obtain the statistics of the entire game and delete the original columns of section data. Finally, we obtain a clean dataset which consists of 10364 rows and 199 columns. To make it concise, the following abbreviations for basketball in-game statistics in Table 1 will be used in this article.

3 FEATURE ENGINEERING

In previous sections, a cleaned and structured dataset is obtained after scraping and preprocessing the data. However, this form of data is not of much use for future predictions because all those in-game statistics will be known only after a game is ended, not before the game, which makes them useless for predicting future game results. Therefore, in this section, some new features that are useful for predictions for future games are created based on the original datasets. In order to make better predictions, not only traditional internal factors such as teams' in-game statistics including PTS, TRB, ORB, DRB, AST and so on, but also external factors that potentially affect the outcome of game such as home court advantage and players' tiredness due to successive games are taken into consideration. The overall framework of designed features is shown in Figure 1

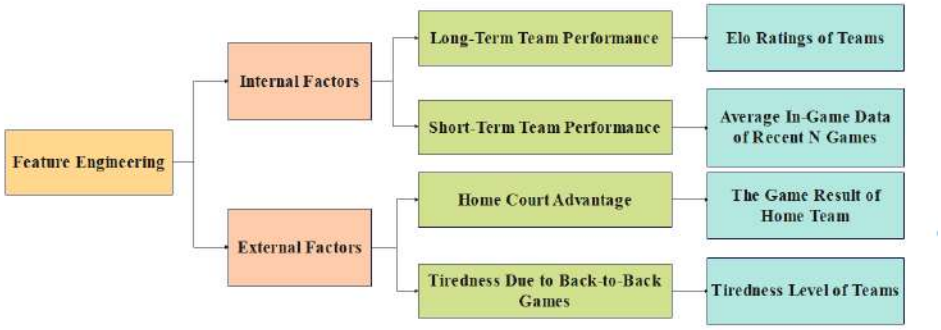


Figure 1: Framework of Feature Engineering Including Both External and Internal Factors

3.1 Internal Factors: Long-Term and Short-Term Performance

Internal factors that determine which team will win the game refer to teams' intrinsic strengths, such as the team's overall abilities of three-point shooting, rebound, block, steal and assist. Considering the fact that competitiveness of teams and players shows both short-term and long-term characteristics, we evaluate the current strength of a team by both its long-term and short-term performance. We utilize the team's Elo rating to represent its long-term performance since the 2012-13 season, and utilize the team's average in-game statistics in the recent N games to represent its short-term performance, where N is a hyperparameter.

3.1.1 Short-Term Team Performance: Average Statistics in Recent N Games. Historical data shows that even within a single season, the intrinsic strength of NBA teams often fluctuate drastically. As a result, a team's performance in recent games becomes an important predictor of its future success. In this article, we use the average performance over the past N games to represent a team's recent strength, where N is a constant. A larger value of N reflects team's performance within a relatively long period, while a smaller N reflects team's performance within a relatively short period, which is more adaptive and sensitive to latest changes. To discover the optimal N that achieves the highest predictive accuracy, we utilize N =3,4,5,. . .,10 to construct 8 sets of features and train models on them respectively. For each N, 38 features are created, 19 features for home team and 19 for away team, which includes the average PTS, TRB, ORB, DRB, AST, TOV, STL, BLK, FT, LFT, FGA, FG, FGM, 2FGA, 2FG, 2FGM, 3FGA, 3FG, 3FGM in recent N games. Note that the calculation of the above features is in fact the process of properly inserting moving averages of team statistics according to chronological order. We first rearrange the dataset by dates of games. For each game in the dataset, we then search upwards in our dataset for the most recent N games that the home team had played and calculate the mean of certain statistics. After that, we do the same for the away team.

3.1.2 Long-Term Team Performance: Elo Ratings. To evaluate the long-term performance of sports players and teams, a widely adopted methodology is the Elo rating [4], a method first proposed by the American physicist Arpad Elo for calculating the relative skill levels of players in zero-sum games such as chess. When applied to sport games such as NBA basketball games, it can measure the long-term performance of teams based on their win-loss record. The basic principle is that the winner team will gain more Elo points from the loser team when defeating stronger rivals or winning by larger margins. Take the game between Phoenix Suns and Golden State Warriors on Nov 30$^{\text{th}}$, 2021 as an example, before the game started, Phoenix Suns has a rating of 1684.838 and Golden State Warriors has a rating of 1664.601. Denote the Suns as team A and Warriors as team B, then the winning probability of team A (Phoenix Suns) is

$$P_{win} = \frac{1}{1 + 10^{\frac{-(Elo_A - Elo_B + hca)}{400}}} = \frac{1}{1 + 10^{\frac{-(1684.838 - 1664.601 + 69)}{400}}} = 0.626$$

where *hca* represents home court advantage which approximately equals to 69 Elo points in previous researches [7]. Since team A (Phoenix Suns) defeated team B (Golden State Warriors) by 8 points (104 to 96), the Elo points that team A received from team B is:

$$\Delta Elo = k \times (1 - P_{win}) = 18.457 \times (1 - 0.626) = 6.909$$

where k is a moving constant that depends on the margin of victory and difference in Elo ratings between two teams in a certain game, which is formulated by:

$$k = 20 \times \frac{(MoV + 3)^{0.8}}{7.5 + 0.006 \times Elo_{diff}} = 18.457$$

where MoV represents the margin of victory which is the points that winner team gets more than loser team, and Elo_{diff} represents the difference of Elo rating between team A and B. It can be easily inferred that a lager MoV and a smaller Elo_{diff} can both result in a lager k, making the transfer of Elo points ΔElo from the winner to the loser team much more significant. Thus, the Elo ratings of two teams after this game are:

$$Elo_A + \Delta Elo = 1684.838 + 6.909 = 1691.747$$

$$Elo_B - \Delta Elo = 1664.601 - 6.909 = 1657.692$$

That is, after this game, the Elo rating of Phoenix Suns was increased to 1691.748, while the one of Golden State Warriors was decreased to 1657.692. Using the same method in this example, the Elo rating of home team and away team before and after each game can be calculated. Note that initial values of the Elo rating of all teams are set to 1500 ($Elo_{Initial}$) at the beginning of the

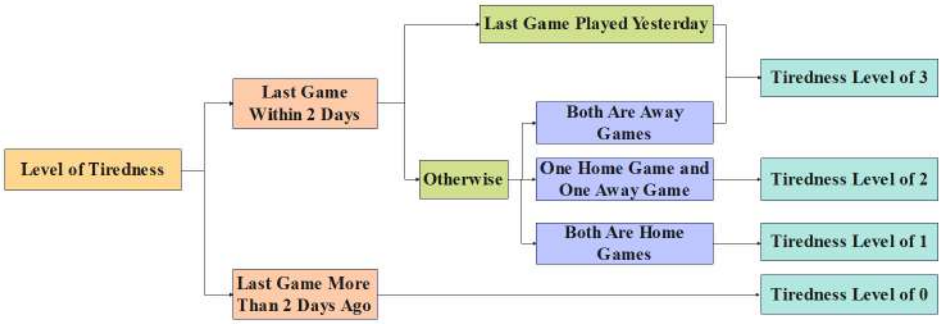


Figure 2: Level of Tiredness Determined by Dates and Locations of Games

2012-13 season. Then, teams' Elo ratings before and after each game from 2012-13 season to 2021-22 season are calculated via an iterative algorithm based on the points of two teams and the location of the game. In order to capture the uncertainty caused by team recruitments between seasons, at the beginning of each season the Elo Rating of each team is reset as:

$$Elo_{New\ Season} = 0.75 \times Elo_{Previous\ Season} + 0.25 \times Elo_{Initial}$$

where $Elo_{Initial}$ is the initial value of the Elo Rating System of NBA and is set to 1500 in this article.

3.2 External Factors: Level of Tiredness and Home Court Advantage

Only utilizing internal factors for prediction inevitably results in inaccuracy because multiple external factors that potentially affect game results are neglected, i.e. tiredness due to back-to-back games, home court advantage and players' injuries. Here we consider tiredness and home court advantage for prediction.

3.2.1 Level of Tiredness. NBA games are often scheduled on Friday and weekend nights to attract more audience. It is common occurrence that teams have to fly to another city for tomorrow night's game right after tonight's game is finished. Games played by the same team consecutively on two days or even within the same day are called back-to-back games, which along with long travels unavoidably cause stress and tiredness of players, thus affecting team performances and even skewing game results. Historical data shows that many times outcomes of games were skewed because teams were playing back-to-back. According to NBC Sports, in 2008-09 season, 22 out of 30 NBA teams had worse records in games which were back-to-back.

In our model, the tiredness of team caused by back-to-back games is categorized to 4 types. If a team played its last game yesterday or played 2 away games within the past 2 days, its tiredness is defined as 3. If a team played one away game and one home game within the past 2 days, its tiredness is defined as 2. If a team played two home games within the past 2 days, its tiredness is defined as 1. However, if a team played its last game more than 2 days ago, the team does not suffer from tiredness and therefore its tiredness is defined as 0. In this way, the tiredness level of home and away team of each game is calculated and added to our dataset as new features.

3.2.2 Home Court Advantage. Home Court Advantage refers to the benefits that the home team gains over the visitor team in sport games. From physiological perspective, playing in familiar environment make teams play better because players do not need to recover from jet lag, adapt to unfamiliar climates and foods. In addition, psychological advantages of home team also exist, for example, the support from the local audience. Previous research found that home court advantage approximately equals to 69 Elo points. Therefore, we add extra 69 Elo points to home team in each game as is shown in formula (1). Besides, labels of models are coded as 1 if home team wins and 0 if it loses, which implicitly distinguish the home and away team so that the model can discover the hidden rules and patterns of home court advantage via learning from past data.

4 EXPOLORATORY DATA ANALYSIS

4.1 Density of Elo Ratings

4 4 shows the kernel density estimation of Elo ratings of 30 NBA teams from 2012-13 season to 2021-22 season. It can be seen that in most seasons, the distribution of Elo ratings subjects to normal distribution. Seasons like the 2013-14 and 2019-20 season contain a more even distribution of teams' strengths, while seasons like the 2012-13 and 2017-18 season shows a more skewed and imbalanced distribution. In seasons where teams have similar strengths, the outcomes of games are more uncertain and difficult to predict, and are likely to rely much more on external factors such as teams' tiredness level and home court advantage.

4.2 Winning Rate by Teams

5 5 shows the accumulative winning rate of 30 NBA Teams over the past 10 regular seasons since 2012-13 season. The most successful team over the past 10 regular seasons is GSW (Gold State Warriors), followed by SAS (San Antonio Spurs), LAC (Los Angeles Clippers), TOR (Toronto Raptors) and OKC (Oklahoma City Thunder). Teams with the most upset winning rate are ORL (Orlando Magic) and MIN (Minnesota Timberwolves).

Besides, it can be inferred that none of the 30 NBA teams dominate the league over the past 10 years because even the strongest team does not achieve an accumulative winning rate over 70 percent in regular seasons. This illustrates that teams in the NBA league

Figure 3: Home Court Advantage

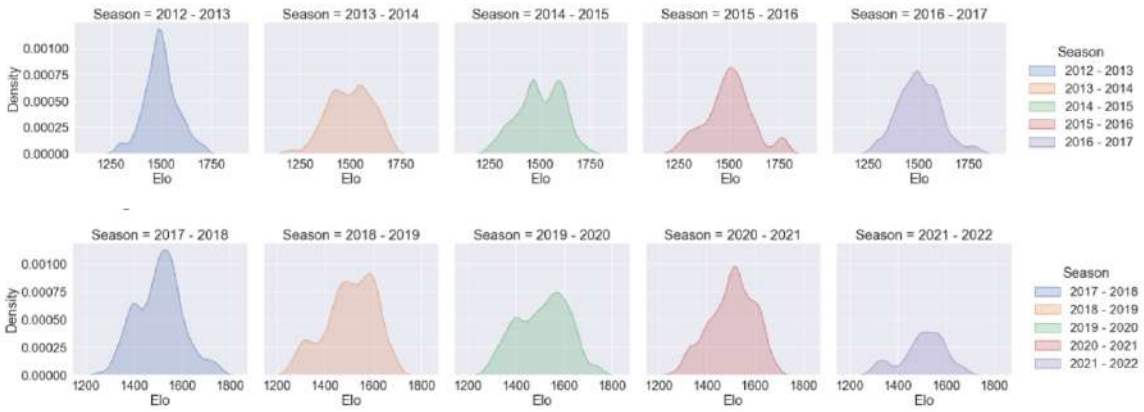


Figure 4: Density of Elo Ratings Per Season

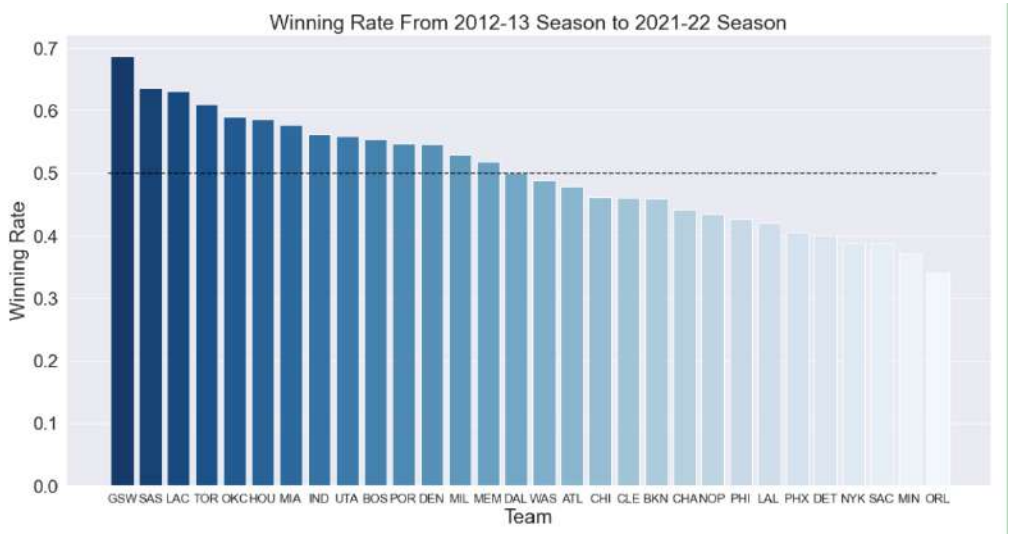


Figure 5: Winning Rate From 2012-13 Season to 2021-22 Season

are very close to one another in terms of team strength and share an even chance to win at most of the time. Thus, predicting the results of NBA games are not a trivial task. A prediction accuracy close to 70 percent is already quite satisfying.

5 BUILDING UP FEATURE SETS

After creating features for future game prediction, we verify the effectiveness of them by training and testing models respectively on three feature sets.

5.1 Feature Set A

Feature Set A contains 38 features of average in-game statistics in recent games, Elo ratings and home court advantage, which is the feature set utilized by past researches and has been proved to be effective. Based on feature set A which is treated as the baseline feature set, we further extend it to feature set B and C by proposing some new features.

5.2 Feature Set B

Feature Set B extends feature set A by adding two tiredness variables for home team and away team, respectively. If models trained on feature set B outperform the ones trained on A, the tiredness features can be proved to be effective.

5.3 Feature Set C

Feature Set C further extends feature set B by adding 21 new features that emphasize the differences between home and away team, which are calculated by home team statistics subtracting the corresponding away team statistics divided by home team statistics. We denote them by D, which stands for differences. For example, we denote Elo rating of home team subtracting Elo rating of away team divided by Elo rating of home team as "D_{Elo}" and average ORB of home team subtracting average ORB of away team divided by average ORB of home team as "D_{ORB}". The motivation of creating these features is to make features more straightforward and understandable to models since they directly reflect to which extent the home team is performing better than the away team. If models trained on feature set C outperforms those trained on A and B, the "D" variables are proved to be effective.

6 FEATURE SELECTION

6.1 Sequential Forward Selection

Sequential Forward Selection (SFS) is an iterative feature selection method where the model starts with the best performing feature against the target, and then select another feature that gives the best performance in combination with the first selected variable. This process continues until the preset stopping condition is satisfied, for example, the number of features to be selected is reached.

6.2 Recursive Feature Elimination

Recursive Feature Elimination (RFE) is an iterative feature selection method which selects features by recursively considering smaller sets of features. To begin with, the estimator is trained on the initial set of features. Then, the least important features are pruned from the current set of features. This procedure is recursively repeated until the preset number of features to be selected is reached.

7 MODEL SELECTION

In this section, we rearrange the dataset by chronological order and split it into training set (90%) and testing set (10%) without shuffling the data. Then, based on three feature sets, different machine learning models are trained using 10-fold cross validation on training set which contains 9161 NBA games from 2012-13 to 2020-21 season. After that, top models are chosen for final evaluation on testing set which contains 1036 games from 2020-21 season to 2021-22 season. After a wide selection of models, the goal is to find out the best predictive model for future games, especially games in the latest 2020-21 and 2021-22 season.

7.1 Feed Forward Neural Network

From Table 2, it can be seen that neural network models trained on feature set B, C generally outperforms those trained on feature set A for each N, demonstrating that new features are helpful for the prediction. When selecting models for further hyperparameter tuning and final evaluation, we take both mean accuracy and standard deviation into consideration. The chosen ones are indicated in bold in Table 2

7.2 SVM

From Table 3, it can be seen that SVM models trained on feature set C generally have much lower standard deviation and relatively high mean accuracy, demonstrating that new features in B and C improve models' performance. Models indicated by bold is selected due to low variance and relatively high predictive accuracy.

7.3 Logistic Regression

Three conclusions can be drawn from Table 4. The first conclusion is that logistic regression models trained on feature set B, C generally outperforms models trained on feature set A. The second conclusion is that feature selection using SFS and RFE improves model performance by both reducing the standard deviation and enhancing the accuracy. And the third is that the SFS significantly outperforms RFE algorithm when implementing feature selection for logistic regression on our datasets.

As shown in Table 4, the best model achieves an accuracy of 67.39% and a standard deviation of 1.11%, which is the logistic regression model with SFS feature selection trained on feature set C using information of the past 3 games. Among 61 features, 21 influential features are selected by the SFS algorithm, including tiredness of home and away team, and 7 "D" features, which verifies the effectiveness of the new features we design.

7.4 Random Forest

From Table 5, we can see that SFS and RFE feature selection do not make significant improvement to random forest models. This may be due to the decision tree, the base learner of random forest, is able to identify and choose important features when deciding where to split during training. Therefore, no external feature selection algorithm is needed any more. It is also observed that random forest models without feature selection generally have a 0.1%-2% lower

Table 2: Cross Validation Results for Feed Forward Neural Network Models

Feature	Model	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Feature Set A	FFNN	66.58%	66.58%	66.76%	66.59%	66.83%	66.78%	66.75%	**66.80%**
		(0.91%)	(0.04%)	(0.20%)	(0.90%)	(0.29%)	(0.86%)	(0.75%)	**(0.15%)**
Feature Set B	FFNN	66.83%	66.75%	66.65%	66.66%	66.71%	66.70%	66.82%	**66.88%**
		(0.76%)	(0.62%)	(0.77%)	(0.44%)	(0.40%)	(0.42%)	(0.38%)	**(0.16%)**
Feature Set C	FFNN	66.90%	67.01%	67.01%	66.80%	66.90%	66.84%	**66.93%**	66.29%
		(1.04%)	(0.23%)	(0.53%)	(0.42%)	(0.79%)	(0.53%)	**(0.09%)**	(1.40%)

*The bold indicates the selected models.

Table 3: Cross Validation Results for SVM Models

Feature	Model	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Feature Set A	SVM	66.77%	**66.81%**	66.65%	66.58%	66.78%	66.81%	66.69%	66.52%
		(2.01%)	**(1.72%)**	(2.17%)	(2.04%)	(2.51%)	(2.94%)	(1.87%)	(1.86%)
Feature Set B	SVM	66.53%	66.58%	66.55%	66.46%	66.67%	66.65%	66.73%	**66.74%**
		(2.72%)	(2.95%)	(2.51%)	(2.26%)	(2.23%)	(1.84%)	(2.52%)	**(0.31%)**
Feature Set C	SVM	66.66%	66.78%	**66.78%**	66.61%	66.72%	66.48%	66.56%	66.62%
		(2.12%)	(1.18%)	**(0.22%)**	(0.42%)	(0.48%)	(0.35%)	(0.58%)	(0.64%)

*The bold indicates the selected models.

Table 4: Cross Validation Results for Logistic Regression Models

Feature Set	Model	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Feature Set A	SFS_LR	67.08%	67.07%	67.11%	67.01%	67.21%	67.09%	67.09%	67.03%
		(1.03%)	(0.85%)	(0.72%)	(0.87%)	(1.18%)	(1.07%)	(1.10%)	(1.21%)
	RFE_LR	66.72%	66.86%	66.87%	66.76%	66.80%	66.66%	66.71%	66.76%
		(2.35%)	(1.69%)	(2.24%)	(2.48%)	(2.26%)	(2.06%)	(2.15%)	(2.23%)
	LR	66.64%	66.86%	66.94%	66.73%	66.79%	66.89%	66.90%	66.85%
		(2.23%)	(2.06%)	(2.12%)	(2.24%)	(2.00%)	(1.85%)	(1.89%)	(1.97%)
Feature Set B	SFS_LR	67.17%	67.10%	67.15%	67.11%	**67.28%**	67.23%	67.21%	67.00%
		(1.03%)	(0.69%)	(0.90%)	(1.02%)	**(1.29%)**	(1.10%)	(1.06%)	(0.80%)
	RFE_LR	66.82%	66.89%	66.86%	66.74%	66.79%	66.87%	66.76%	66.83%
		(2.59%)	(1.96%)	(2.53%)	(2.35%)	(2.19%)	(2.16%)	(2.43%)	(2.45%)
	LR	66.79%	66.86%	66.98%	66.78%	66.78%	66.89%	66.85%	66.74%
		(2.50%)	(1.90%)	(2.50%)	(2.16%)	(2.85%)	(2.30%)	(2.26%)	(2.23%)
Feature Set C	SFS_LR	**67.39%**	67.35%	67.35%	67.13%	67.30%	67.14%	67.25%	67.16%
		(1.11%)	(1.03%)	(1.17%)	(1.10%)	(0.94%)	(1.15%)	(1.24%)	(1.11%)
	RFE_LR	66.88%	66.86%	66.89%	66.73%	66.80%	66.77%	66.85%	66.96%
		(2.47%)	(2.19%)	(1.87%)	(1.90%)	(1.80%)	(2.17%)	(1.97%)	(2.10%)
	LR	66.84%	66.78%	66.91%	66.67%	66.77%	66.84%	**67.04%**	66.92%
		(2.04%)	(2.00%)	(2.52%)	(1.78%)	(1.99%)	(1.92%)	**(2.19%)**	(2.02%)

*The bold indicates the selected models.

standard deviation compared with other machine learning models trained on the same datasets which is due to the nature of the random forest algorithm ——it aggregates multiple high variance and low bias decision trees together to reduce variance and enhance overall performance.

7.5 Naïve Bayes

From Table 6, it can be seen that naïve bayes models with SFS feature selection outperform models without feature selection for game data from 2012-13 to 2020-21 NBA season. The best model achieves a mean accuracy of 67.48% with a low standard deviation of 1.03%, which is the naïve bayes model with SFS feature selection trained on Feature Set C using information over the past 5 games. Among 61 features, 15 influential features are selected by the SFS,

Table 5: Cross Validation Results for Random Forest Models

Feature Set	Model	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Feature Set A	SFS_RF	66.39%	66.64%	66.58%	66.39%	66.87%	66.76%	**67.12%**	66.86%
		(1.10%)	(1.05%)	(1.17%)	(1.21%)	(1.15%)	(1.29%)	**(1.07%)**	(1.63%)
	RFE_RF	65.88%	66.10%	65.98%	65.81%	65.98%	66.20%	66.41%	66.23%
		(3.20%)	(2.07%)	(2.21%)	(2.60%)	(2.51%)	(2.40%)	(3.17%)	(2.37%)
	RF	66.66%	66.75%	66.72%	66.90%	67.07%	66.89%	66.83%	66.85%
		(2.33%)	(2.11%)	(1.99%)	(1.45%)	(1.73%)	(1.92%)	(1.91%)	(2.01%)
Feature Set B	SFS_RF	66.78%	66.66%	66.41%	66.48%	66.26%	66.41%	66.65%	66.90%
		(1.02%)	(1.09%)	(1.58%)	(1.06%)	(0.29%)	(0.40%)	(1.34%)	(1.28%)
	RFE_RF	65.74%	66.08%	66.14%	66.06%	66.15%	65.93%	66.08%	66.37%
		(3.39%)	(3.42%)	(1.84%)	(3.47%)	(1.68%)	(2.67%)	(3.47%)	(2.14%)
	RF	66.58%	66.68%	66.76%	66.94%	66.97%	66.88%	66.89%	66.84%
		(2.10%)	(1.91%)	(1.68%)	(1.43%)	(2.05%)	(1.95%)	(1.88%)	(1.99%)
Feature Set C	SFS_RF	66.88%	66.65%	66.67%	66.61%	66.42%	66.68%	66.51%	66.89%
		(1.60%)	(1.29%)	(0.03%)	(0.12%)	(0.48%)	(1.29%)	(0.41%)	(0.22%)
	RFE_RF	66.35%	66.22%	66.28%	66.24%	66.50%	66.39%	66.22%	66.48%
		(1.77%)	(2.33%)	(2.33%)	(2.03%)	(2.69%)	(2.80%)	(2.27%)	(2.82%)
	RF	66.53%	66.86%	**67.13%**	66.71%	66.79%	66.73%	66.61%	**66.73%**
		(1.17%)	(0.60%)	**(0.41%)**	(0.04%)	(0.04%)	(0.18%)	(0.47%)	**(0.05%)**

*The bold indicates the selected models.

Table 6: Cross Validation results for Naïve Bayes Models

Feature Set	Model	N=3	N=4	N=5	N=6	N=7	N=8	N=9	N=10
Feature Set A	SFS_NB	67.06%	67.30%	67.18%	67.05%	67.14%	67.14%	67.15%	67.10%
		(1.03%)	(1.21%)	(0.63%)	(1.15%)	(1.50%)	(1.20%)	(0.94%))	(0.83%)
	RFE_NB	\	\	\	\	\	\	\	\
	NB	66.82%	66.99%	66.88%	66.82%	66.97%	67.01%	66.91%	67.06%
		(2.34%)	(1.83%)	(1.91%)	(1.84%)	(2.39%)	(2.31%)	(2.63%)	(2.89%)
Feature Set B	SFS_NB	67.05%	67.16%	67.18%	67.02%	67.24%	67.17%	67.15%	67.10%
		(0.91%)	(1.20%)	(0.63%)	(1.27%)	(1.32%)	(1.34%)	(0.94%)	(0.83%)
	RFE_NB	\	\	\	\	\	\	\	\
	NB	66.82%	67.01%	66.88%	66.83%	66.96%	67.01%	66.90%	**67.06%**
		(2.36%)	(1.82%)	(1.91%)	(1.86%)	(2.41%)	(2.31%)	(2.38%)	**(2.71%)**
Feature Set C	SFS_NB	67.18%	67.30%	**67.48%**	67.16%	67.32%	67.26%	67.04%	67.26%
		(1.14%)	(0.98%)	**(1.03%)**	(1.30%)	(1.49%)	(1.31%)	(1.11%)	(1.05%)
	RFE_NB	\	\	\	\	\	\	\	\
	NB	66.59%	66.61%	**66.60%**	66.60%	66.59%	66.54%	66.55%	66.56%
		(2.17%)	(2.17%)	**(2.02%)**	(2.15%)	(2.01%)	(2.04%)	(2.23%)	(2.27%)

*RFE cannot be implemented on naïve bayes classifier since the algorithm does not have feature importance. The bold indicates the selected models.

including tiredness of home team, tiredness of away team, and 5 "D" features (tiredness, Elo rating, average 2FG, 2FGA and TOV), which indicates the effectiveness of the new features in feature set B and C.

8 FINAL EVALUATION

8.1 Fair Prediction

It is worth noting that our prediction is "fair" because we do not shuffle the dataset and stratify it by label when splitting the training and testing set. Instead, based on our 10 season's data, we utilize the first 90% data to train and select models, and then forecast game

Table 7: Fair Prediction Results of the latest 2021-22 Season

Model	Feature Set	N	Cross Validation Accuracy(Standard Deviation)	Testing Accuracy
FFNN	Feature Set A	N=10	66.80%(0.15%)	**66.50%**
FFNN	Feature Set B	N=10	66.88%(0.16%)	**67.58%**
FFNN	Feature Set C	N=9	66.93%(0.09%)	64.73%
SVM	Feature Set A	N=4	66.81%(1.70%)	64.63%
SVM	Feature Set B	N=10	66.74%(0.31%)	**67.39%**
SVM	Feature Set C	N=5	66.78%(0.20%)	64.05%
SFS_LR	Feature Set B	N=7	67.28%(1.29%)	64.44%
SFS_LR	Feature Set C	N=3	67.39% (1.11%)	64.44%
LR	Feature Set C	N=9	67.04%(2.19%)	64.73%
SFS_RF	Feature Set A	N=9	67.12%(1.07%)	**65.32%**
RF	Feature Set C	N=5	67.13%(0.41%)	64.93%
RF	Feature Set C	N=10	66.73%(0.05%)	**67.98%**
NB	Feature Set B	N=10	67.06% (2.71%)	**67.88%**
NB	Feature Set C	N=5	66.60%(2.02%)	63.16%
SFS_NB	Feature Set C	N=5	67.48% (1.03%)	63.85%

*The bold indicates accuracy above 65% in the testing set.

Table 8: Top Five Models

Model	Feature Set	N	Test Accuracy	AUC
FFNN	Feature Set A	N=10	66.50%	60.40%
FFNN	Feature Set B	N=10	67.58%	62.57%
NB	Feature Set B	N=10	67.88%	63.88%
SVM	Feature Set B	N=10	67.39%	63.06%
RF	Feature Set C	N=10	67.98%	63.96%

results of the 10% data in the latest 2020-21 and 2021-22 season, which may decline our accuracy but makes the results more useful and meaningful in real world application.

8.2 Prediction Results of the Latest 2020-21 and 2021-22 NBA Season

Based on previous analysis, 15 models are selected to conduct further hyperparameter tuning. Models with optimal hyperparameters are tested on our testing set which contains 1036 games in the latest 2020-21 and 2021-22 season. The results are shown in 7 7, where several conclusions can be drawn: First, most of the models trained on feature subsets selected by SFS algorithm do not perform as well as models trained on complete feature sets on our testing set, which is probably due to the information loss caused by feature selection. Moreover, random forest and naïve bayes are the best models for the NBA prediction task.

To highlight best models, models that give an accuracy of over 65% on testing set are indicated by bold text in 7 7 and displayed in detail in 8 8 and 6 6. It is indicated that N=10 is the most suitable choice for NBA prediction. Our best model, that is, the random forest model trained on feature set C using information over the past 10 games, achieves the highest prediction accuracy of 67.98% for the latest 2020-21 and 2021-22 NBA season, indicating the features we propose are effective predictors for future NBA games.

9 CONCLUSION

This research discovers useful new features that can be used to make better predictions for NBA basketball games, including level of tiredness and the so-called "D" variables which emphasizes the difference of statistics between home and away team. We started by calculating Elo ratings, creating features based on information in the most recent N games that a team played, and designed the tiredness variables based on dates and locations of games. After that, we built up three feature sets and implemented feature selection using SFS and RFE algorithms with the aim of verifying the effectiveness of the new features and improving models' overall performance. Next, 10-fold cross validation is utilized to train and validate different models for a more robust model selection result. Eventually, we picked 15 top models to conduct further hyperparameter tuning and evaluate their performances on the testing set containing 1036 games in the latest 2020-21 and 2021-22 season. Our best model, which is the random forest model trained on feature set C using information in the past 10 games gives the optimal accuracy of 67.98%, which is quite a decent performance since the highest accumulative winning rate of NBA teams in the regular season is below 70%. Core findings of this research are summed up as follows:

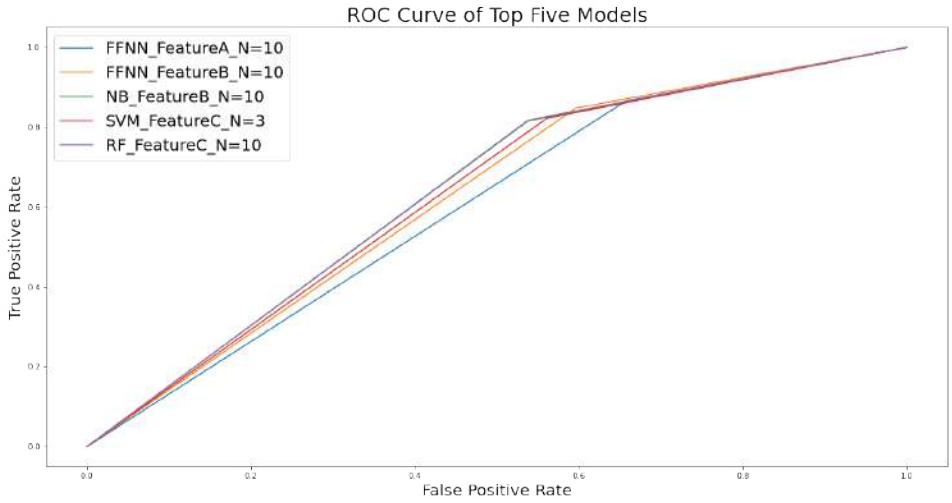


Figure 6: ROC Curve of the Top Five Models

(1) New features including level of tiredness and "D" features improve prediction accuracy, making our best models achieve accuracy that is very close to 68%.

(2) Average in-game statistics over the past 10 games achieves the best predictive performance.

(3) Feature selection improves model performance on training set by both enhancing mean accuracy and decreasing standard deviation, but worsen models' prediction accuracy when applied to new seasons.

(4) Random forest and naïve bayes models are the best models for NBA prediction among all the candidates in this article.

10 LIMITATIONS AND FUTURE WORK

Based on what we have completed in this research study, more improvements can be made. For example, more factors that potentially affect the outcome of games, both internal and external ones, can be taken into consideration. Since this study only considers team level statistics, other internal factors in future studies may include players' individual efficiencies based on player level statistics, such as the position that a player played and his abilities of shooting three-points field goals, rebounding, stealing, blocking and assisting. Besides, except for home court advantage and tiredness of teams, other external factors in future studies may also include injury, resting and return of players. Moreover, more instances can be collected to further improve prediction accuracy which may involve not only regular season games, but also playoffs and even pre-season training games. In addition, apart from the models utilized in this research, a wider collection of candidate models including both ML (machine learning) and DL (deep learning) models can be tested to identity the most effective ones for fair prediction of sports' outcomes, especially for basketball in future studies.

REFERENCES

[1] Cao C (2012) Sports data mining technology used in basketball outcome prediction. Dublin Institute of Technology.
[2] Thabtah, F., Zhang, L., & Abdelhamid, N. (2019). NBA game result prediction using feature analysis and machine learning. Annals of Data Science, 6(1), 103-116.
[3] Nguyen, N. H., Nguyen, D. T. A., Ma, B., & Hu, J. (2021). The application of machine learning and deep learning in sport: predicting NBA players' performance and popularity. Journal of Information and Telecommunication, 1-19.
[4] Hvattum, L. M., & Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. International Journal of forecasting, 26(3), 460-470.
[5] https://projects.fivethirtyeight.com/2018-nba-predictions/games/
[6] http://www.nbaminer.com/
[7] Ferrario, A. BASKETBALL ANALYTICS: THE USE OF DATA SCIENCE TO DESCRIBE AND PREDICT THE PERFORMANCE OF AN NBA TEAM.
[8] Chen, W. J., Jhou, M. J., Lee, T. S., & Lu, C. J. (2021). Hybrid Basketball Game Outcome Prediction Model by Integrating Data Mining Methods for the National Basketball Association. Entropy, 23(4),477
[9] Loeffelholz, B., Bednar, E., & Bauer, K. W. (2009). Predicting NBA games using neural networks. Journal of Quantitative Analysis in Sports, 5(1).
[10] Pai, P. F., ChangLiao, L. H., & Lin, K. P. (2017). Analyzing basketball games by a support vector machines with decision tree model. Neural Computing and Applications, 28(12), 4159-4167.

A Data-Driven Analysis of Formula 1 Car Races Outcome

Ankur Patil[1], Nishtha Jain[2], Rahul Agrahari[2], Murhaf Hossari[2], Fabrizio Orlandi[2], and Soumyabrata Dev[2,3(✉)]

[1] National College of Ireland, Dublin, Ireland
[2] ADAPT SFI Research Centre, Dublin, Ireland
soumyabrata.dev@ucd.ie
[3] School of Computer Science, University College Dublin, Dublin, Ireland

Abstract. There are a range of factors that affect the outcome of Formula 1 (F1) car races. Today, it is reasonable to say that F1 races are first won at the factory, and then on the track. F1 teams accumulate enormous amounts of data during races. In this paper, we propose a data-driven approach to identify the most important factors that contribute to the overall points scored by each driver in a F1 season. We perform a correlation analysis along with a principal components analysis (PCA) to identify the factors that are closely related. Furthermore, using PCA, we efficiently reduce our 21 input variables into a lower-dimensional subspace, that can explain most of the variance in our data and which is easier to comprehend. We obtain 5 years (2015–2019) of data explaining the F1 car characteristics from a publicly available website https://www.racefans.net/. We use this web-scrapped F1 race study to understand the impact of the different car features on the total points scored by a driver in the season. To the best of our knowledge, our work is the first of its kind in the area of F1 car races.

Keywords: Formula-1 · Feature analysis · Data analytics · Open-source code

1 Introduction

One of the most popular sports in the world is Formula 1 (F1). The speed thrill and nail-biting experience that fans get while watching the race is the result of a lot of engineering, data science, management, and of course lots of training on the tracks. What is often underappreciated is that races are won first at the factory and then on the circuit. The F1 teams work hard to maintain a constant balance between obtaining top speed and down force, here aerodynamics plays a major role [12]. The teams try to predict what position they will finish by using the massive datasets they have accumulated from the past seasons. It would be worthwhile to dig deeper into such a sport and analyze the associated analytics

A. Patil and N. Jain—Authors contributed equally.

L. Longo and R. O'Reilly (Eds.): AICS 2022, CCIS 1662, pp. 134–146, 2023.
https://doi.org/10.1007/978-3-031-26438-2_11

to understand its impact on the total car race points accumulated by a F1 driver. In this work, we provide a data-driven framework to understand the various car race statistics, and check their impact on the performance of the F1 racers.

There are 4 basic strategies that the driver/team uses during the race. This will assist us in understanding the basic fundamentals of F1 race, before diving deep into the winning prediction methodologies.

- **Preparation**: The engineering team works on developing a strategy which is based on simulations and data that have been acquired from the various trial runs the driver takes and also based on the past races.
- **Practice**: The driver practices and uses the strategy provided. This is a great way to correct the shortcomings in the strategy. This acts as a stepping stone in fine-tuning the strategy for the qualifying and final race.
- **Qualifying**: The driver moves on to taking the qualifying rounds and the starting position. The previous practice and qualifying rounds feed very critical information to the engineering team to work on the final race's pit stops and race strategy.
- **Racing**: Technical difficulties are a part and parcel of F1 races, but there a few other things that act as catalyst to the victory or loss of the team/driver. Weather conditions, traffic on the tracks, pit stops for tyre or oil changes or other quick fixes, and of course the safety car which limits the speeding cars from crashing when obstructions appear on the track.

Now that we are aware of the 4 important stages in devising a successful strategy for winning the race, it is time to dive deeper into our objective of improving the decision-making steps using statistical tools and techniques.

1.1 Related Work

In the literature, a lot of work is done by researchers in predicting the different sports results using machine learning techniques. In the work by Bishell [4], the author performed experiments on horse races data, and implemented a neural network for predicting the horse race outcomes. Bishell concludes that a simple neural network model gave more efficient and accurate results compared to other benchmarking models. The neural model managed to achieve an accuracy of 66% for the top three ranks. Another research conducted by William and Li [19] using the data collected from Caymans Race Track in Jamaica. The authors implemented the model by using a neural network and achieved an overall accuracy of 74% to predict the top three positions. Similar research was done in 2010 by Dacoodi and Khanteymoori [7], they acquired the data from Aqueduct Race Track in NY. Their work proposed a neural net that has an accuracy of 77%, when compared with other neural networks. In [13], Miljković *et al.* did research on predicting the outcomes of basketball matches using Naive Bayes, using the data acquired from the NBA website. The model achieved an accuracy of 77.97%. Also, [9] predicted the outcomes of the matches played by Tottenham Hotspurs in English Premier League. In [14], the author proposes a model developed from

Bayes networks to predict the expert knowledge in the game of football. The author concludes that Bayes network achieved an accuracy of 59.21% and outperforms other benchmarking algorithms. Recently, in 2017, [18] conducted a research analysis on predicting the football matches played in English Premier League using Bayes Networks and the prediction accuracy was 75.09% on an average across three seasons.

The majority of this prior research focused on developing predictive models with high generalization accuracy (as measured by performance on test sets) rather than on analyzing the factors that contribute to the outcome of a sports event. Furthermore, F1 races and its related analysis have been largely ignored. To the best of our knowledge, there is no publicly published work that provides a systematic analysis on race features that influences the outcome in F1 races. In this paper, we attempt to bridge this gap and provide a detailed analysis on F1 car analytics.

1.2 Contributions of This Paper

The main contributions of this paper are as follows:

- Firstly, we propose a novel and systematic analysis of the various F1 car race factors in our collected dataset, that govern the finishing position of a driver and the manner in which they are related to each other;
- Secondly, we successfully reduced the data space comprising 21 race features into 4 orthogonal dimensions that explain approximately 70% of the captured variance, using principal components analysis. This will facilitate us in identifying the key factors of F1 car race influencing the race outcome;
- Finally, in the spirit of reproducible research, we release all the code and associated dataset with this work. The data set for this domain of sports analytics is a bit difficult to obtain in the form of direct `CSV` files. We have web scraped the data using `R` language for this work. We subsequently converted this data set from multiple pages on the website into a re-usable `CSV` file.

The rest of the paper is organized as follows. Section 2 discusses the various factors associated with a F1 car race. Section 3 describes their inter-dependency in details. We perform a dimensional reduction of the original feature space, using PCA in Sect. 4. Subsequently, we analyze the impact of the different car race features in total race points in Sect. 5. Finally, Sect. 6 concludes the paper and discusses the future works.

2 Formula-1 (F1) Car Race Factors

In this section a brief discussion is done on data collection, data pre-processing and transformation of the input data.

2.1 Dataset

The dataset used in this paper has been acquired from a single source. This dataset is obtained after web scrapping using R studio. With the spirit of reproducible research, the dataset and code for this work is reproducible and is available online[1]. The dataset was taken from https://www.racefans.net/2018-f1-season/2018-f1-statistics/. We collected the data for a period of 5 years (2015–2019).

The dataset provides information on the following attributes:

- Average number of pit stops taken by each racer across the board is represented by `Average.Pit.Stop`
- Information about % usage of each tyre type is represented by variables `Hard`, `Medium`, `Soft`, `Super.soft`, `Ultra.soft`, `Hyper.soft`, `Wet` and `Intermediate` - which denote their % use
- Laps each driver spent in each position during the season considering only first, second and third position is represented by the variables `FirstPosition`, `SecondPosition` and `ThirdPosition`
- # of races the driver started is represented by the variable `Started`, # of races the driver classified by completing 90% of the race is represented by the variable `Classified` and # of races the driver completed by covering 100% race distance in the season is represented by the variable `Completed`
- Full season laps led (represented by `Full.seasons.laps.led`) and driver's season laps led (represented by `Driver.s.season.laps.led`) explain the number of laps led as percentage during the season and all race laps covered by that driver respectively
- # of accidents by each racer in the season is represented by the variable `Accident`
- # of penalties attained to the team and driver for each driver are represented by `Penalties.due.to.team` and `Penalties.due.to.driver` respectively. Simultaneously, if there was no penalty given, it counts as a no action and that is represented by the variable `No.action`
- Average position where each driver started every race, after penalties were applied is represented by the `Average.pole.position`
- The total number of points scored by the driver during the season is denoted by `Total.Points`. These points eventually decide the winner of each season

2.2 Data Pre-processing

This section gives us insights on how missing values, data transformation and data pruning were dealt with in order to carry the analysis forward.

Data Pruning: Data pruning refers to getting rid of unwanted data which are not required for analysis. In our case we performed data pruning on the attributes which were outliers and had no significance on the analysis. The

[1] https://github.com/nshthj/F1_race_exploratory_analysis.

attribute `Withdrawn` (W) described all the drivers who had withdrawn from the race. Here all the racers did participate and there was no driver who withdrew. So, this attribute was removed. Also the attribute `Did Not Qualify` (DNQ) consists all the data for racers who did not qualify. However, all the drivers did qualify for the final race and hence this attribute was removed.

Handling the Missing Values: After data pruning missing values were detected and analyzed as to why they are absent. The missing values in the data is not because of faulty data entry or avoided data. It is because the driver has not been involved in that event. As an illustration, in the event of an accident, only a couple of drivers were affected. Hence, the missing values were replaced with zero.

2.3 Data Transformation

The variables that underwent transformation are as follows:

- Pit Stop data was mentioned according to each lap i.e. 22 laps. A mathematical average was calculated and average pit stop for each driver was created.
- Tyres data was in the form of a percentage. All the special characters were taken off and the percentage was normalized to a decimal format.
- Full season laps led and Driver's season laps led was in the form of percentage which was normalized to a decimal format.

The variables what we have in the dataset are all considered to be important. However, there are 22 variables so having a feature selection process in place to get more independent and uncorrelated input variable set becomes all the more important. Most classification algorithms thrive on input variables that are independent of each other in order to explain maximum variation and trends in the dataset. This paper essentially explores these different variable selection processes. We first talk about a rather straightforward correlation analysis and then move on to a more comprehensive principal components analysis.

3 Interdependency of Variables

In this section, we do a correlation analysis [5,6] of all the variables described in the aforementioned sections. We have used the R function `corrgram`[2]. In our case, as mentioned all the attributes are considered important for the research and there was no manual removal of features. It is important to understand the correlation trend [1,17] between the different features before we perform any classification task. This is because if two features are perfectly correlated, then one feature can be efficiently described by the other [11,16]. Figure 1 depicts how attributes are correlated with each other.

[2] https://www.rdocumentation.org/packages/corrgram/versions/1.13/topics/corrgram.

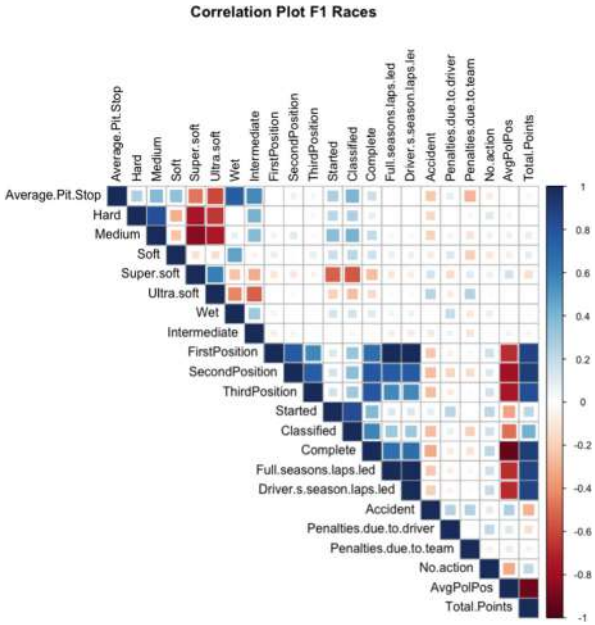


Fig. 1. Correlation between the various F1 car race variables (best viewed in color).

We observe that the average pole position is strongly negatively correlated with the first, second and third position. This makes sense as a higher average pole position would perhaps mean the racer didn't finish in the first, second or third position at the end of the race - also depicting that the average pole position is perhaps one of the key factors in determining the finishing position of the driver. Interestingly, we observe that team penalties appear to be related to the usage of soft tyres and hyper soft tyres – using soft tyres more often generate less penalties while usage of hyper soft tyres will generate more penalties. Moreover, hyper soft tyres are positively correlated with the occurrence of accidents - in line with the fact that they can cause more penalties. Additionally, the position features viz. first, second and third position are strongly positively dependent on the number of laps completed, the full seasons laps led by the drivers and whether the driver was classified or not. Another interesting relationship is the strong negative correlation between a driver classifying and the occurrence of accidents.

4 Principal Components Analysis

In addition to the inter-dependency of the different variables, we also use Principal Component Analysis (PCA) [3,15] to understand the underlying structure of the dataset. Let us assume that our F1 race features are the column vectors $\mathbf{v}_{1-22}$ (22 in our case), where $\mathbf{v}_j \in \mathbb{R}^{n \times 1}$ where $j = 1, 2, \ldots, 22$, and n is the

total number of observations in the dataset. We stack the individual feature vectors $\mathbf{v}_j$ to create the variable matrix $\mathbf{X} \in \mathbb{R}^{n \times 22}$:

$$\mathbf{X} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{22}]. \tag{1}$$

We normalize each of the feature vectors $\mathbf{v}_j$ with the corresponding mean value $\bar{v_j}$ and the standard deviation σ_{v_j} to compute the normalised matrix $\ddot{\mathbf{X}}$. We compute the matrix $\ddot{\mathbf{X}}$ as:

$$\ddot{\mathbf{X}} = \left[\frac{\mathbf{v}_1 - \bar{v_1}}{\sigma_{v_1}}, \frac{\mathbf{v}_2 - \bar{v_2}}{\sigma_{v_2}}, .., \frac{\mathbf{v}_j - \bar{v_j}}{\sigma_{v_j}}, .., \frac{\mathbf{v}_{22} - \bar{v_{22}}}{\sigma_{v_{22}}} \right]. \tag{2}$$

We thereby compute the covariance matrix of $\ddot{\mathbf{X}}$. Subsequently, we perform eigenvalue decomposition of the computed covariance matrix to obtain the eigen values and the eigen vectors. The eigen values describe the amount of variance captured by each of the principal components. The principal components are obtained from the eigen vectors.

4.1 Variation Explained by the Components

In this section, we analyze the variance captured by the most important principal components. Figure 2 describes the variance captured by each of the *orthogonal* principal components. We observe that the first two principal components capture 50% of the total variance. Furthermore, the cumulative variance captured by the first 4 principal components is $\approx 70\%$. This indicates that most of the race features are correlated with each other (as observed in Sect. 3), and the total information in the original feature space can be effectively reduced to a lower dimensional subspace without the loss of significant information.

4.2 Bi-plot Representation

We also represent the car race variables in the new subspace representation of the principal components. Figure 3 is the bi-plot representation [2,8] of our race variables across the first two principal components in a two-dimensional space. We represent the different race observations in our dataset by points in the bi-plot figure. We represent the race car variables by vectors. The bi-plot figure provides us interesting insights on the F1 car race variables. We can observe the contribution of each of the race variables onto the principal components, and also the correlation between them. The position variables viz. `FirstPosition`, `SecondPosition`, `ThirdPosition` are correlated with each other and have a strong contribution to the second principal component. In addition to that, other variables related to the driver's position in the race are quite strongly contributing to PC1 - thus making it a PC that potentially explains the positional aspect of the driver. We also observe that accident and penalties due to team are correlated with each other. We don't see a similar dependence of variables on any other PCs, hence the other three components explain the variation in the input variables in a cumulative manner.

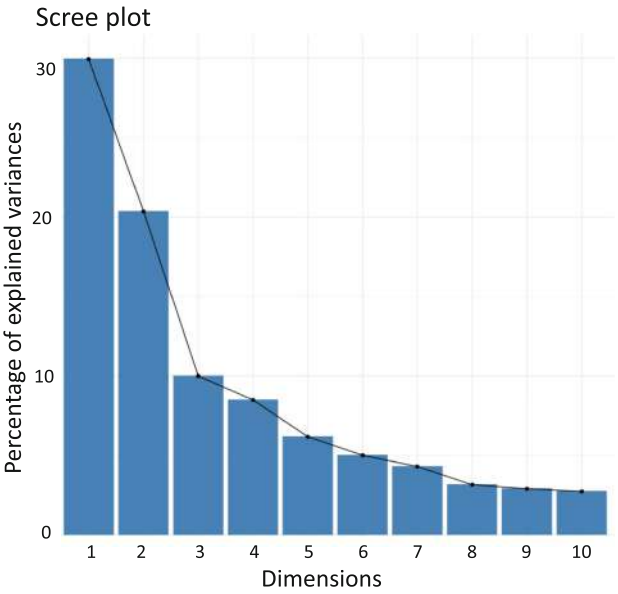


Fig. 2. Amount of variance captured by the individual principal components.

4.3 PCA Factor Loadings

The PCA factor loadings explain the loading that each variable has on each of the components. It also shows the range of loadings on each principal component from each variable [10]. Table 1 describes the loading factors of the various car race features onto the first four principal components. The bold loadings show the top 6 loading *magnitude-wise* on each principal component. It helps us understand what could each principal component potentially represent. For example, similar to the findings in the previous section, the first PC shows strong loadings for all position-related variables. Similarly, the third PC has maximum loadings on the tyre related variables, thus accounting for the variance based on the type of tyre used during the race. It is also possible for one variable to have high loadings on multiple principal components, as can be seen in the table as well.

5 Impact on Season's Total Championship Points

We have discussed the relationship between the different factors that determine the final race outcomes. In this section, we run a linear regression on the data obtained from web-scraping. This data consists of information from 5 consecutive seasons of 2015 till 2019. The dependent variable in the linear regression is the total points scored by a driver in each season denoted by `Total.Points`. This is chosen as the dependent variable, because eventually the driver with

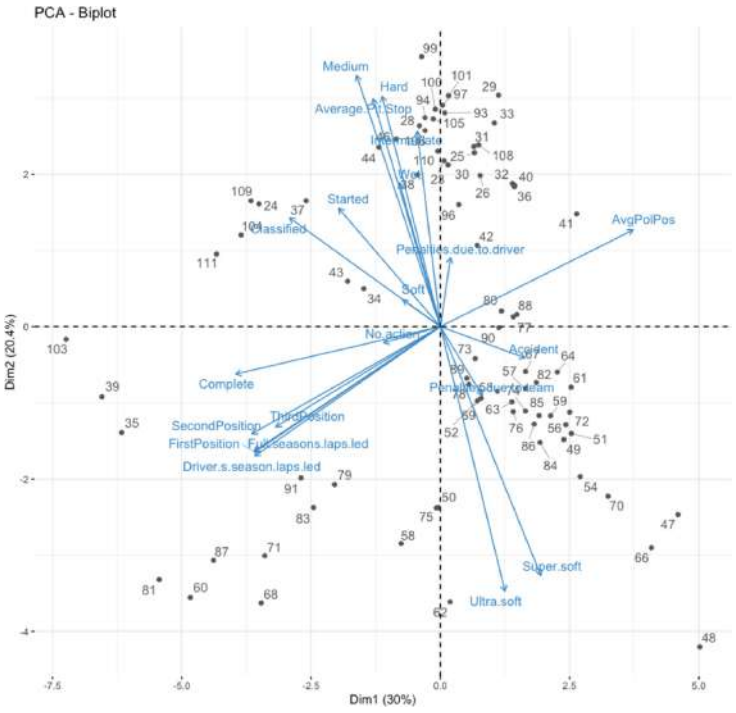


Fig. 3. Biplot representation of the F1 race variables across the first two principal components. The F1 variables are represented by the vectors and the observations in the dataset are represented as points.

the highest points wins the season. We propose to study the effect of our input variables on `Total.Points`. In Table 2, we show the results of a linear regression model that was applied on our dataset. We can observe that number of races completed by a driver (`Complete`) in a season has a significance effect on `Total.Points`. In addition to that, for every race that a driver completes in a season, `Total.Points` increases by 6 units. We also observe that, amongst all the tyre types, only `Medium`, `Soft`, `Ultra.Soft` and `Intermediate` tyre types have a significant effect on `Total.Points`. According to the linear regression results, for a percentage increase in `Intermediate` during the season, the `Total.Points` increases by 4. We also observe that a percentage increase in the use of `Medium`, `Soft` and `Ultra.Soft` tyre types (which are also the most used tyre types in the season), the total points scored increase by 2 for each. In addition to these, an increase in the number of laps spent by the driver in second position, denoted by `SecondPosition`, the `Total.Points` will increase by 0.20. The results are similar for `ThirdPosition`. An interesting finding of this model is also the effect of `Average.Pol.Pos` on `Total.Points`. The feature `Average.Pol.Pos` denotes the average starting position held by each driver during the course of the season. A unit increase in the `Average.Pol.Pos` will result in a decrease of 3 points

Table 1. Loading factors of the various features onto the first four principal components.

Race features	PC_1	PC_2	PC_3	PC_4
Average.Pit.Stop	−0.118	**0.330**	**−0.346**	−0.128
Hard	−0.102	**0.333**	**0.387**	−0.085
Medium	−0.147	**0.363**	0.313	−0.038
Soft	−0.066	0.039	**−0.550**	0.094
Super.soft	0.177	**−0.360**	−0.137	−0.110
Ultra.soft	0.114	**−0.383**	0.004	0.189
Wet	−0.072	0.208	**−0.487**	−0.040
Intermediate	−0.041	0.283	−0.021	−0.245
FirstPosition	**−0.328**	−0.181	0.003	−0.076
SecondPosition	**−0.331**	−0.155	0.026	−0.123
ThirdPosition	−0.289	−0.145	−0.010	−0.048
Started	−0.179	0.171	−0.009	**0.520**
Classified	−0.265	0.157	−0.040	0.305
Complete	**−0.360**	−0.069	−0.022	0.036
Full.seasons.laps.led	**−0.327**	−0.182	−0.001	−0.076
Driver.s.season.laps.led	**−0.326**	−0.187	−0.001	−0.065
Accident	0.149	−0.045	0.009	**0.388**
Penalties.due.to.driver	0.018	0.100	−0.125	**0.413**
Penalties.due.to.team	0.074	−0.100	0.172	0.162
No.action	−0.100	−0.023	0.153	0.327
Average.Pol.Pos	**0.339**	0.140	−0.041	−0.063

in `Total.Points`. The linear regression model has an R-squared value of 99% which means that the model was able to capture almost 99% of the variation in the data.

6 Conclusion and Future Work

In this paper, we have provided a systematic analysis of various variables associated with the F1 car race. We have identified the most important variables that assist in a favorable outcome of the car race. Using a set of statistical techniques, we concluded that most of the variables are strongly correlated with each other. We also surmised that the original feature space can be significantly reduced to a lower-dimensional subspace without a significant loss of information.

Future work include extending such systematic analysis for a larger statistical period of more than 5 years to gather more data and investigate the analysis further. Furthermore, we plan to investigate the linear regression model by

Table 2. We show the corresponding estimate and p-value for all the car race features, while estimating the total race points accumulated by a driver in a complete season. The significance codes are represented by '+', where 0: '+++', 0.001: '++', 0.01: '+', 0.05: '.', and 0.1: ''.

Race features	Estimate	p-value	Signif. codes
Average.Pit.Stop	−11.29	0.37	
Hard	1.09	0.35	
Medium	2.11	0.10	
Soft	2.13	0.08	
Super.soft	1.69	0.17	
Ultra.soft	2.11	0.09	
Wet	1.28	0.43	
Intermediate	3.62	0.05	+
FirstPosition	0.21	0.73	
SecondPosition	0.20	7.34e−07	+++
ThirdPosition	0.19	6.53e−07	+++
Started	−0.68	0.62	
Classified	0.03	0.99	
Complete	5.56	7.96e−05	+++
Full.seasons.laps.led	−2.15	0.82	
Driver.s.season.laps.led	3.39	0.30	
Accident	0.26	0.91	
Penalties.due.to.driver	−1.24	0.29	
Penalties.due.to.team	1.45	0.13	
No.action	0.44	0.77	
Average.Pol.Pos	−3.44	0.02	+

modifying it to use a selected set of race features by applying forward and/or backward step regression.

Acknowledgement. This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at University College Dublin. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme. The authors would also like to thank Prof John D. Kelleher from Technological University Dublin, Ireland for helpful discussions on this work.

References

1. Alparslan, B., Jain, M., Wu, J., Dev, S.: Analyzing air pollutant concentrations in New Delhi, India. In: 2021 Photonics & Electromagnetics Research Symposium (PIERS), pp. 1191–1197. IEEE (2021)
2. AlSkaif, T., Dev, S., Visser, L., Hossari, M., van Sark, W.: A systematic analysis of meteorological variables for PV output power estimation. Renew. Energy **153**, 12–22 (2020)
3. Batra, S., et al.: DMCNet: diversified model combination network for understanding engagement from video screengrabs. Syst. Soft Comput. **4**, 200039 (2022)
4. Bishell, A.: Machine learning and New Zealand horse racing prediction. BSc. Report, Department of Computer Science, Massey University, New Zealand (2006)
5. Danesi, N., Jain, M., Lee, Y.H., Dev, S.: Monitoring atmospheric pollutants from ground-based observations. In: 2021 IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), pp. 98–99. IEEE (2021)
6. Danesi, N., Jain, M., Lee, Y.H., Dev, S.: Predicting ground-based PM2.5 concentration in Queensland, Australia. In: 2021 Photonics & Electromagnetics Research Symposium (PIERS), pp. 1183–1190. IEEE (2021)
7. Davoodi, E., Khanteymoori, A.R.: Horse racing prediction using artificial neural networks. Recent Adv. Neural Netw. Fuzzy Syst. Evol. Comput. **2010**, 155–160 (2010)
8. Dev, S., Lee, Y.H., Winkler, S.: Color-based segmentation of sky/cloud images from ground-based cameras. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **10**(1), 231–242 (2017)
9. Joseph, A., Fenton, N.E., Neil, M.: Predicting football results using Bayesian nets and other machine learning techniques. Knowl.-Based Syst. **19**(7), 544–553 (2006)
10. Manandhar, S., Dev, S., Lee, Y.H., Winkler, S., Meng, Y.S.: Systematic study of weather variables for rainfall detection. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 3027–3030. IEEE (2018)
11. Manandhar, S., Dev, S., Lee, Y.H., Meng, Y.S., Winkler, S.: A data-driven approach for accurate rainfall prediction. IEEE Trans. Geosci. Remote Sens. **57**(11), 9323–9331 (2019)
12. Martins, D., Correia, J., Silva, A.: The influence of front wing pressure distribution on wheel wake aerodynamics of a F1 car. Energies **14**(15), 4421 (2021)
13. Miljković, D., Gajić, L., Kovačević, A., Konjović, Z.: The use of data mining for basketball matches outcomes prediction. In: Proceedings of IEEE 8th International Symposium on Intelligent Systems and Informatics, pp. 309–312. IEEE (2010)
14. Pariath, R., Shah, S., Surve, A., Mittal, J.: Player performance prediction in football game. In: Proceedings of Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1148–1153. IEEE (2018)
15. Pathan, M.S., Nag, A., Dev, S.: Efficient rainfall prediction using a dimensionality reduction method. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 6737–6740. IEEE (2022)
16. Pathan, M.S., Nag, A., Pathan, M.M., Dev, S.: Analyzing the impact of feature selection on the accuracy of heart disease prediction. Healthc. Anal. **2**, 100060 (2022)
17. Pathan, M.S., Wu, J., Lee, Y.H., Yan, J., Dev, S.: Analyzing the impact of meteorological parameters on rainfall prediction. In: Proceedings of IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), pp. 100–101. IEEE (2021)

18. Razali, N., Mustapha, A., Yatim, F.A., Ab Aziz, R.: Predicting football matches results using Bayesian networks for English Premier League (EPL). In: Proceedings of IOP Conference Series: Materials Science and Engineering, vol. 226, p. 012099. IOP Publishing (2017)
19. Williams, J., Li, Y.: A case study using neural networks algorithms: horse racing predictions in Jamaica. In: Proceedings of International Conference on Artificial Intelligence (ICAI 2008), pp. 16–22. CSREA Press (2008)

Predicting Formula 1 Race Outcomes: A Machine Learning Approach

Ali Jafri
Computer Science, NYUAD
aj3218@nyu.edu

Advised by: Djellel Difallah, Talal Rahwan

ABSTRACT

This project explores the application of machine learning to predict Formula 1 lap times and race outcomes, leveraging historical data spanning from 2014 to 2023. The primary objective is to develop a predictive model capable of accurately forecasting lap times for individual drivers throughout a race. By utilizing advanced machine learning techniques, particularly Long Short-Term Memory (LSTM) networks, this research aims to uncover the complex relationships between various factors influencing lap times, such as track characteristics, team performance, driver performance, and historical trends.

The study begins with a comparative analysis of simpler models like linear regression, decision trees, and random forests to establish baseline performances. LSTM models are then employed to handle sequential data and capture temporal dependencies in lap times. The results demonstrate the effectiveness of LSTMs in predicting relative driver performance and race outcomes, with refinements such as custom loss functions improving accuracy. Findings from preliminary tests revealed that minimizing laptime loss alone does not always translate to better race outcome predictions.

The evaluation includes testing on the 2023 Abu Dhabi Grand Prix and 2024 Bahrain and Abu Dhabi Grands Prix, showcasing significant improvements in prediction accuracy with later model iterations. Future work will explore transformer models for their ability to capture global dependencies and extend prediction scope to include pit stop strategies and safety car laps. This research not only advances predictive analytics in motorsport but also offers valuable insights into Formula 1 dynamics, enhancing engagement for teams, broadcasters, and spectators alike.

KEYWORDS

Formula 1, machine learning, motorsport, lap analysis, laptime prediction race prediction, LSTM, transformers, regression

Reference Format:
Ali Jafri. 2024. Predicting Formula 1 Race Outcomes: A Machine Learning Approach. In *NYUAD Capstone Seminar Reports, Spring 2024, Abu Dhabi, UAE.* 10 pages.

1 INTRODUCTION

Formula 1 racing stands at the pinnacle of motorsport, where drivers push the limits of human and machine performance on some of the world's most challenging circuits. Central to the competitive landscape of Formula 1 is the quest for optimal lap times, which directly influence race outcomes and hence affect championship standings. In this context, the prediction of lap times emerges as a critical endeavor, offering insights into driver performance, race strategies, and ultimately, the determination of race winners [1, 4].

The importance of accurately predicting Formula 1 lap times cannot be overstated. In the high-stakes environment of such elite motorsport, by forecasting lap times, teams gain a strategic edge in race planning, pit stop optimization, and tire management, all of which are crucial factors in securing victory. Moreover, for broadcasters and spectators, lap time predictions add an extra layer of excitement and engagement, enabling informed speculation and analysis throughout the race. Having a tool that can assist with making such educated guesses not only demystifies such a complex sport but can also deepen the sense of involvement for all audiences.

To address the challenge of lap time prediction in Formula 1, this research builds upon a foundation of data-driven analysis and machine learning methodologies. Drawing from extensive datasets spanning multiple seasons and race circuits, the project aims to uncover the complex relationships between various factors influencing lap times, including track

This report is submitted to NYUAD's capstone repository in fulfillment of NYUAD's Computer Science major graduation requirements.

جامعة نيويورك ابوظبي
NYU ABU DHABI

Capstone Seminar, Spring 2024, Abu Dhabi, UAE

characteristics, car performance, and driver behavior. Therefore the main research question of this project is 'How accurately can we predict lap times of a given race using machine learning methodologies?'.

However, the task of predicting Formula 1 lap times presents several formidable challenges. The dynamic nature of racing environments, the interplay of multiple variables, and the inherent uncertainty of such a competitive sport pose significant obstacles to accurate forecasting. Yet, it is precisely these challenges that make this work both hard and exciting. Success in this endeavor promises not only advancements in predictive analytics but also deeper insights into the intricate dynamics of Formula 1 racing, perhaps even pushing the boundaries of what is achievable in sports analytics and machine learning.

2 RELATED WORK

2.1 Machine learning analysis

An important utilization of machine learning techniques is to find the importance of certain variables for determining a particular outcome. That is exactly what one paper has done, by using tree-based models to find the relevance of variables such as starting position, constructor points, and driver points in terms of finishing top 3 in a given race [12]. Another study that was done had a similar approach, where instead of using tree-based models, it utilized LSTMs to find explanatory variables. The findings of this paper were essentially that the performance of a driver in the qualifying session is crucial to determining the end position of said driver in a race [13]. The work done by both of these papers will be useful for this project as it will allow the choosing of the correct variables for the dataset, and make sure that any key variables are not left out.

2.2 Predicting or Optimizing Strategy

Along with the analysis of what variables are important, another use of machine learning in this context is to find optimal solutions in order to maximize performance. In this case, machine learning could be used to determine optimal pit stop strategies. This is exactly what is proposed in a Master's thesis, whereby the author aims to automate the identification of tire strategies for Formula 1 races by treating the problem as a sequential decision-making process. What is interesting about this approach is that the paper designs a planning environment that replicates past Formula 1 races using a lap time simulator based on regression techniques applied to publicly available race data [10]. This is quite interesting as it allows one to see how making a certain decision for a certain driver can change the whole race's landscape. Another paper used a slightly different approach, which was to use a predictive model for determining the optimal timing of pit stops during Formula 1 races. The author utilized 3 different machine learning algorithms (Support Vector Machines (SVM), Random Forest, and Artificial Neural Networks) to determine which one would be best in not only predicting whether a driver makes a pitstop at a given lap of the race, but whether the pitstop was a 'Good Pitstop' [8]. Such an approach would definitely be of use when trying to create a predictive model that not only takes into account raw driver performance but also the pitstop strategies of different teams.

2.3 Predicting race outcomes

Perhaps one of the most important applications of machine learning in the realm of Formula 1 is the prediction of race outcomes. One paper does exactly that by trying to predict the winner of Formula 1 races using Python and Support Vector Machines. Their model was trained using historical data from F1 races, such as lap times, sector times, qualification times, and information about the drivers and teams. They set their experiment as a classification problem, where they would determine whether a driver would finish on the podium, finish in the points (top 10 on the grid) or would not finish in the points (placed 11th to 20th position on the grid) [11].

Another paper that tries to predict race outcomes uses generalized additive models (GAM) to represent the evolution of lap times in Formula 1 races. Their aim is to analyze Formula 1 team and driver performances during a specific race by modeling lap times as a function of relevant predictor variables, both numeric and categorical. The study focuses on the Formula 1 season of 2015 and utilizes freely available data to fit the model. Their results indicate that the model accurately describes race development in Grand Prix events without unpredictable occurrences such as safety car interventions or race suspensions [6]. Additionally, their model shows potential for specifying alternative race strategies, particularly concerning pit stop choices. The approach specified in the paper is especially valuable to the current project, as it predicts specific laptimes rather than general results (as was seen in the previous paper). However, their data pool does not consider the 8-9 additional years of racing that happened after 2015, meaning that a newer model has the potential to be even more accurate.

3 METHODOLOGY

3.1 Gathering the dataset

The first step in this project was to gather all the necessary data required to train the models. There are multiple APIs available that can be used to source this data. One of them being the Ergast Developer API. This API has the lap times for

all drivers for all races in a given season. While the API also has some other data available, like the number of pitstops, it doesn't provide the full details such as what tire compound was on before and after the stop. Another popular API is FastF1, which rather than providing a XML or JSON response (like how the Ergast API usually does) provides a Python library where all the data can be accessed from there. This API offers more than the Ergast API, as not only does it have lap times, but also has the tire strategies used by all drivers in a given race and the laps which were held under a yellow flag or safety car. However, the data for the FastF1 API is limited in that there is no tire compound data before 2018. For additional data, such as finding the number of wins of a driver up to a given point in the season, the StatsF1 website was used. Using all of these APIs and alternate methods will result in a raw dataset that includes data from 2014 all the way up to 2023.

The dataset comprised information from 203 races, 55 unique drivers, and a total of 214607 laps. Each row gives you information on a lap completed by a certain driver in a given race. The dataset can be split into three different categories: core features, race conditions, and driver and team flags. The core features consist of things like race and driver identifiers (the ID of the race, driver, constructor, etc.), performance metrics (grid position and qualifying lap times), and driver statistics (number of total wins, races completed, podiums, points this season, etc.). The race conditions consist of safety car indicators (whether a safety car was deployed during or before a lap) and pit stop information (whether the driver is going to pit this lap, the compound of the tire, and the age of the tire). The driver and team flags (in the format of isHAM to signify that this is Lewis Hamilton's lap, or isRBR to signify that this lap is of a driver who races for Red Bull) indicate which driver or team the lap corresponds to. The target variable is the time (in milliseconds) that the driver took to complete that lap. This is the primary outcome that the model aims to predict using the features mentioned above.

3.2 Comparative and Reproducibility study

In this section, we delve into the comparative analysis of various machine learning models and their applicability to predicting Formula 1 race outcomes. This analysis involves understanding the strengths and weaknesses of different models, reproducing results from existing projects, and discussing performance outcomes. The selection of models for this project is based on their potential to handle the complexities of Formula 1 data. The models considered include: linear regression, logistic regression, decision tree models, random forest models, LSTMs, and transformers.

For linear regression, it was seen that it was suitable for predicting continuous variables like lap times. While this is easy to interpret and fast to train, it assumes linear relationships which may not capture the intricate dynamics of racing data. Nevertheless, this could still be a potential candidate for predicting the lap times of the drivers for a race. Logistic regression was looked at for its potential with binary classification tasks like predicting whether a driver pits, whether there is a safety car, or (on a more macro-scale) whether a driver finishes on the podium [9]. A potential good use for this is also multi-class predictions, such as what tire (out of the three compounds available) a driver will put on when they come in for a pitstop. However, since the main focus of this project is to predict lap times, the maximum that classification models can do is act as a supplementary layer that predicts safety cars, pitstops, and tire compound usage. This predicted information can then be used to predict lap times.

Decision tree models are also a good potential option as they can do both regression and classification tasks. Moreover, they can capture complex relationships (where linear regression and logistic regression may not be able to). However, based on research, it seems that decision tree models are quite prone to overfitting, especially with noisy data [7]. This is where random forest models might be better, as they can still utilize decision tree models' ability to capture complex relationships, but can reduce overfitting by averaging over multiple trees. However, it would be harder to interpret the model and understand the importance of certain features of the dataset in predicting the desired outcomes.

Long Short-Term Memory Networks (also known as LSTMs) are another potential candidate for predicting Formula 1 lap times. These models are ideal for time-series prediction and capturing long-term dependencies in sequential data like lap times. Since lap times are sequential and depend on previous laps' conditions, LSTMs can be effective by learning the temporal relationships. Moreover, this model type has been used in other projects for very similar use cases (Jared Chan, 2021). Another potential model type is transformers, which use time-series prediction (like LSTMs) but are more powerful in capturing global dependencies over long sequences. However, for these model types (both LSTMs and transformers) the complexity of setting up the model and training the models is significantly higher compared to the previous models mentioned. Additionally, transformers specifically are known to work well with particularly long sequences [14] and for the use case that this project explores, we are limited to only 50 to 70 laps for a given race.

Now that we have our dataset and also an understanding of the models that can be used, we shall now try and reproduce the results of two existing projects with the dataset that was made for this project. Starting with Veronica Nigro's project,

we can see that her aim is to predict race outcomes like who wins the race. She used data from 1983 to 2018 (using the 2019 season as the testing data) and included things like race information, results, weather, driver and team standings, and qualifying times. She had originally looked into logistic and linear regressions, random forests, support vector machines, and neural networks from both a regression and classification perspective. Upon doing some testing and tuning she found that the neural networks classifier model was the best suited and gave the best results, correctly predicting the winner for 62% of the races in 2019. Using the same tuning parameters but with a slightly modified version of the dataset meant for this project, we were able to get a score of 90% for the 2019 season (training the model only on races from 2014-2018). If the tuning parameters were modified slightly (changing hidden layer size and alpha value) we got an accuracy of 86% for the 2022 season (using up to the 2021 season as training only). For 2023, (training upto 2022) we got a score of 95%. Although this seems quite high, it must be noted that the 2023 season would have been very predictable given that Max Verstappen had won almost all of the races. Nevertheless, we have seen from our results of 2019 that the model that was trained using the dataset for this project performed significantly better (even with the same training parameters as the final model by Veronica Nigro). Hence, we can say that even if the approaches to training are similar, in the end, the dataset plays a crucial role in determining the model's performance.

The second project we look at is by Jared Chan and uses LSTMs to predict lap times, positions, and pit stop strategies for up to 20 drivers throughout a race. He uses data from 2001 to 2019, using 2020 as the testing data, and has an interesting structure for his dataset, whereby each record in his data for a given race contains the lap information for all 20 drivers. This information for each driver includes the driver's ID, the driver's standing, the constructor's standing, the driver's status, the driver's race position, whether the driver is pitting, and the lap time (the last three being the target variables that are going to be predicted by the model). To reproduce these results with the dataset meant for this project meant that a lot of modifications would be needed not only the data we had, but also how it would be used by Chan's code for training and evaluating. When evaluating his model, it received an average loss score of 34.34, while the model trained using the restructured version of our dataset got a loss score of 39.35. While the units are arbitrary as the data was scaled and averaged between the three target variables, we can see that the model that used our dataset doesn't perform as well as the original project's model. A potential reason for this is that the setup of his model and helper functions must have been optimized for the structure of his dataset. Our dataset has a bit more variables per driver (for example the driver statistics which contain the number of wins a driver has), and we were also not focusing on the status variable of the laps, which might have played a negative role. Moreover, when seeing the actual predictions from the model using our dataset, there were some outlandish predictions. For example, we were getting results that Latifi (a driver who is known to barely ever finish in the top 10 and is usually part of the bottom half of the grid [5]) was getting podiums. So results like this might not seem outlandish to the model, but this is a huge mistake to anyone with any knowledge of F1 context.

Based on the findings from the comparative and reproducibility study, it is evident that focusing on LSTMs (Long Short-Term Memory networks) and using relatively simpler models (linear regression, decision trees, or random forests) as baselines is the best route for this project. The primary reason for focusing on LSTMs lies in their ability to handle sequential data effectively, which is a key characteristic of Formula 1 lap times. Unlike simpler models like linear regression, which struggle with capturing complex temporal dependencies, LSTMs excel in modeling time-series data by learning relationships across laps. The reasonable success of Jared Chan's project further reinforces the potential of LSTMs for this use case, as his model demonstrated adequate performance in predicting lap times, positions, and pit stop strategies by leveraging temporal patterns in race data. While reproducing his results using our dataset revealed some challenges, such as differences in dataset structure and outlier predictions, these issues highlight the importance of tailoring the model to the specific characteristics of our dataset. We have also seen from Veronica Nigro's project that how much of an increase in performance can happen just by changing the dataset. Although LSTMs have been explored in similar contexts before, their application using this enriched dataset provides a novel opportunity to fully realize their potential in capturing the complex dynamics of Formula 1 racing. By leveraging the unique features of our dataset—such as detailed driver statistics and tire strategies—this project has the opportunity to push the boundaries of what LSTMs can achieve in Formula 1 lap time prediction.

3.3 Setting up the baseline model

To choose the baseline model we first need to find out which model (out of linear regression, decision trees, and random forests) performs the best using the default parameters set by the Sci-kit Learn package. The dataset used was split into training and test data, with 2014 to 2022 being the training data and the 2023 season being the test data. The data was also scaled using the StandardScaler function in the Sci-kit Learn package in order to standardize the input data such that the data points have a balanced scale. The inputs were the same structure as the original dataset (where each row

gives you information on a lap completed by a certain driver in a given race) and hence used all the core features, race conditions, and driver and team flags. The output was the lap time in milliseconds for that particular lap. Each of the models were trained and then tested on the 2023 season data, getting the necessary scores in the form of mean absolute error, mean squared error, root mean squared error, and the r-squared score.

3.4 Setting up the LSTM model

For the initial setup, we had used the original structure of the dataset without any modifications, and scaled the features and the target using the StandardScaler by Sci-kit Learn. Once scaled, the data was arranged in sequences such that each sequence item was a list of the laps completed by a certain driver in a certain race. The targets (the lap times) were also arranged ina similar array, where each item was the list of the lap times for a certain driver in a certain race. Then these were put on PyTorch tensors and padded to ensure consistent lengths. This step was important because not all the races are of the same length as some can be 44 laps while others can be 78. Then a dataset class was initialized with the sequence and target tensors and would return a particular driver's sequence of laps, lap times, and the length of their laps sequence (essentially the number of laps completed in a given race). The length is important for packing the padded sequence during training so that the model is not trained on the padding values. PyTorch's DataLoader was used to load batches of sequences during training. For the loss function, PyTorch's mean squared error function (MSELoss) was used. The models at this stage were trained with a hidden layer size of 128, 1 LSTM layer, no dropout, a learning rate of 0.001, a DataLoader batch size of 32, and only 10 epochs.

After a few training iterations and experimenting with different parameters, several improvements were made to how the training was done. For example, although 10 epochs were sufficient for Jared Chan's project, it was later realized that 10 epochs were too low for this project. This could be due to the fact that we are dealing with a smaller range of data compared to the other project. Thus, it was increased to 50 epochs with an early stopping mechanism added with a patience value of 10. This meant that if after 10 epochs the loss value hasn't gone down, then the model may have reached its plateau in terms of performance. Spending additional epochs for training might cause overfitting and be a waste of computational resources. Additionally, dropout was introduced to ensure that no overfitting occurred and the model can generalize better. Moreover, more LSTM layers were added to increase the trained model's capability to represent complex functions and relationships in the data.

Another important modification was to employ a manual scaler instead of the StandardScaler. This had multiple benefits, one of which was that it made the loss output interpretable. When using the StandardScaler we would get a training loss value of (for the sake of example) 0.26 and a test loss (the loss seen during testing the model on the 2023 season data) of 0.07. While this may be useful for a relative context where we see how the loss goes up or down based on changing parameters, it doesn't really give an idea of how many seconds off the mark are the predicted lap times compared to the actual lap times. While with the custom scaler, we know exactly the units of the loss output giving a good idea of how well the model performs in more absolute terms. Another benefit of using a custom scaler is that the StandardScaler calculates scaling based on mean and standard deviation, which can be influenced by outliers in the dataset. The custom scaler avoids this issue by using fixed scaling factors instead of relying on the statistical properties of the data.

Additionally, the loss function was modified from simply using PyTorch's mean squared error function to a custom loss function that combines lap time loss, position loss, and historical loss. The lap time loss is essentially the same as the original loss, where we get the mean squared error between the actual and predicted lap times. Position loss calculates the relative positions of drivers using the predicted lap time sequences in a batch, and compares them with the actual positions (calculated with the actual lap time sequences). This was introduced because it was seen that the test loss (which used only the mean squared error for the lap time loss) was around 6400 seconds, giving a root mean square error of about 80 seconds. While this isn't the mean absolute error and is probably significantly inflated in comparison due to the squaring of outliers, we have a general idea of the magnitude of the loss. Keeping in mind that the difference between drivers per lap is usually less than 4 seconds [2], having a root mean square error of about 80 seconds means that the lap time predictions may not be able to accurately reflect the relative performance of the drivers within a race, hence highlighting the importance of position loss. The last component of the combined loss function was the historical loss, which penalizes predictions deviating from historical performance expectations. The historical performance expectations are derived from the driver's season points, total wins, and total podiums. This was introduced because during preliminary testing it was often seen that poor-performing drivers were somehow being predicted to reach podiums. For instance, it was seen that Logan Sargeant, a driver who is part of a team that is struggling and is known to basically always finish outside of the top 10 [3], was somehow being predicted as a winner. In contrast, Max Verstappen (a four-time world champion) was being predicted to finish outside the points.

Having a weighted average between lap time loss, position loss, and historical loss (such that the weight distribution of the individual losses can be a tuning parameter) ensures that along with lap times, relative and historical performance is also taken into account.

3.5 Evaluation Methods

To get a deeper understanding of how the model performs it would be useful to see its predictions for certain races. For this, it would be ideal to choose a circuit that doesn't have as many incidents on track and is relatively easier to predict. These track incidents can range from weather changes (meaning that drivers and teams have to adapt their strategy for the wet climate mid-race), tire punctures, collisions, or even DNFs. Based on previous analysis[15], the Abu Dhabi Grand Prix is the race with the lowest average DNFs per season. Hence it makes sense to view the predictions of the 2023 Abu Dhabi Grand Prix to visualize the models' performance. To analyze the performance of the models for 2024, we will be looking at the Bahrain Grand Prix as well as it is also one of the circuits with the lowest average DNFs per season.

When viewing the predictions we shall see how the predicted final race position compares to the actual race positions. The race position is not a separate prediction output but rather inferred from the lap times themselves. The sum of the lap times for an individual driver would give the total race time, which can then be sorted in ascending order for all drivers to find out who finished the race in what order. Hence the total race time is directly linked with the final race positions. In order to compare the predicted positions with the actual positions, we can use four different metrics. The first three metrics are how many of the top 10, top 5, and top 3 in the predicted and actual positions are the same (irrespective of order). The fourth metric is whether the predicted and actual positions have the same winner. If we were to take order into account, this would be quite an unrealistic expectation for the model to predict the exact order for the top 10 correctly. Therefore we have four degrees of accuracy, where the first three are quite flexible as they don't consider order.

4 RESULTS

Now that we have seen how the models were set up, we can start by seeing the results from the linear regression, decision tree, and random forest models.

What we can see from Table 1 is that the random forest model performed the best by all metrics and hence will be used as a baseline to compare the LSTM models. Interestingly, we see that the mean absolute error is about 9.4 seconds (the original units in the table are set to milliseconds). So while this may not seem like a lot, if we reiterate the point of drivers only being about 3 seconds apart from each other in a race, the presence of this magnitude of an error might mean that the model fails to capture the relative performance of the drivers (especially since these models were trained with default parameters and without any custom loss functions).

Table 2.1 shows the predicted positions from various models compared to the actual positions of drivers for the 2023 Abu Dhabi Grand Prix. The models include the baseline random forest model and three LSTM-based models. LSTM model v1 signifies that the models were trained on data from 2014 to 2022 and tested on 2023 data. The v1.1 and v1.2 signify that they were trained using different configurations (different hidden layer sizes, different weights for the combined loss function, etc.). The results are then evaluated based on key metrics such as whether the correct winner was predicted, the number of podium matches, top 5 matches, and top 10 matches. What we can see is that firstly, none of the models correctly predicted Verstappen as the winner, highlighting a limitation in capturing dominant driver performances accurately. Additionally, some of the outlier predictions for each model have been highlighted. We can see that the baseline had quite a few outliers, as it predicted Sargeant winning the race and Zhou performing well, while Verstappen and Perez finish towards the bottom of the pack. The LSTM models v1.1 and v1.2 also had an outlier of predicting Sargeant too high, with v1.2 also predicting him winning the race. The LSTM model with the lowest test loss (which was just the lap time loss) also predicted Sargeant winning the race while Piastri finishing towards the bottom. If we look at Table 2.2, we can see that the LSTM models v1.1 and v1.2 performed significantly better than the baseline model. We can also see that the model with the lowest test loss (which may imply that this might have the most accurate predictions) performs worse than models v1.1 and v1.2. This means that a lower lap time loss does not always translate to better race outcome predictions, even though the final race outcomes are measured using lap times (by summing all the lap times together to get total race time). This goes back to the earlier point about how just focusing on lap time loss might overlook the drivers' relative performance within a race.

After conducting the tests for the 2023 season, the LSTM models were then trained from 2014 to 2023, and then tested on the 2024 season. The same analysis was then done on the 2024 Bahrain Grand Prix, which can be seen in Tables 3.1 and 3.2. Immediately, we can see that the predictions are significantly better and are quite accurate. Especially for models v2.1 and v2.2, the predictions for the top 5 positions not only match but also follow the exact order as the actual positions. We also again see that the model with the lowest test loss or lap time loss was not the one that performed the best. For reference, the 'v2' signifies that the models were trained on data from 2014 to 2023.

Model Type	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R-squared Score
Linear Regression	12695.6254	5776411599.7532	76002.7078	0.0912
Decision Tree	10099.0927	5880952590.9297	76687.3692	0.0747
Random Forest	9437.2788	5066568167.7366	71179.8298	0.2028

Table 1: Performance metrics of different models for lap time prediction in the 2023 season

Position	Actual Results	Predicted Results (baseline)	Predicted Results (LSTM model v1.1)	Predicted Results (LSTM model v1.2)	Predicted Results (LSTM model v1 with lowest test loss for 2023)
1	Verstappen	Sargeant	Leclerc	Sargeant	Sargeant
2	Leclerc	Leclerc	Verstappen	Verstappen	Alonso
3	Russell	Piastri	Norris	Leclerc	Hamilton
4	Perez	Alonso	Piastri	Perez	Perez
5	Norris	Zhou	Perez	Russell	Russell
6	Piastri	Hulkenberg	Alonso	Norris	Verstappen
7	Alonso	Ocon	Russell	Alonso	Leclerc
8	Tsunoda	Gasly	Hamilton	Tsunoda	Ocon
9	Hamilton	Hamilton	Sargeant	Hamilton	Norris
10	Stroll	Norris	Tsunoda	Piastri	Ricciardo
11	Ricciardo	Russell	Ocon	Stroll	Stroll
12	Ocon	Ricciardo	Albon	Zhou	Tsunoda
13	Gasly	Tsunoda	Stroll	Albon	Hulkenberg
14	Albon	Stroll	Zhou	Ocon	Gasly
15	Hulkenberg	Verstappen	Hulkenberg	Hulkenberg	Piastri
16	Sargeant	Albon	Gasly	Ricciardo	Zhou
17	Zhou	Perez	Ricciardo	Gasly	Albon

Table 2.1: Comparison of actual results and predicted results for the 2023 Abu Dhabi Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
Baseline Model (random forest)	False	1	1	5
LSTM Model v1.1	False	2	4	9
LSTM Model v1.2	False	2	4	9
LSTM Model (v1 with lowest test loss for 2023)	False	0	2	7

Table 2.2: Performance comparison of different models for the 2023 Abu Dhabi Grand Prix.

For the results of 2024 Abu Dhabi Grand Prix, when we look at Tables 4.1 and 4.2 we can see that the results don't seem to be as accurate as the predictions for the 2024 Bahrain Grand Prix. One possible explanation is that there were significant incidents that happened at the start of the 2024 Abu Dhabi Grand Prix. For instance, Piastri and Verstappen were involved in a contact that caused them to spin out of control (but still managed to continue racing). Perez also spun due to contact but DNF'ed. Then later Piastri was involved in a minor contact with Colapinto. Due to these incidents, a lot of penalties were also handed out, causing the lap and race times of the affected drivers to further increase [3]. With this in mind, we can see that the models performed fairly

Position	Actual Results	Predicted Results (LSTM Model v2.1)	Predicted Results (LSTM Model v2.2)	Predicted Results (LSTM model with lowest test loss for 2024)
1	Verstappen	Verstappen	Verstappen	Verstappen
2	Perez	Perez	Perez	Alonso
3	Sainz	Sainz	Sainz	Stroll
4	Leclerc	Leclerc	Leclerc	Russell
5	Russell	Russell	Russell	Norris
6	Norris	Stroll	Alonso	Leclerc
7	Hamilton	Alonso	Stroll	Perez
8	Piastri	Norris	Norris	Sainz
9	Alonso	Piastri	Hamilton	Piastri
10	Stroll	Hamilton	Piastri	Hamilton

Table 3.1: Comparison of actual results and predicted results for the 2024 Bahrain Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
LSTM Model v2.1	True	3	5	10
LSTM Model v2.2	True	3	5	10
LSTM Model (with lowest test loss for 2024)	True	1	2	10

Table 3.2: Performance comparison of different models for the 2024 Bahrain Grand Prix.

Position	Actual Positions	Predicted Positions (LSTM Model v2.3)	Predicted Positions (LSTM Model v2.4)
1	Norris	Norris	Hamilton
2	Sainz	Sainz	Norris
3	Leclerc	Leclerc	Russell
4	Hamilton	Verstappen	Sainz
5	Russell	Hamilton	Leclerc
6	Verstappen	Piastri	Verstappen
7	Gasly	Russell	Piastri
8	Hulkenberg	Gasly	Gasly
9	Alonso	Alonso	Alonso
10	Piastri	Perez	Perez

Table 4.1: Comparison of actual results and predicted results for the 2024 Abu Dhabi Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
LSTM Model v2.3	True	3	4	9
LSTM Model v2.4	False	1	5	9

Table 4.2: Performance comparison of different models for the 2024 Abu Dhabi Grand Prix.

well, with v2.3 still managing to predict the winner and the podium in the exact same order as the actual positions.

5 CONCLUSION

This project demonstrates the potential of machine learning in predicting Formula 1 lap times and race outcomes, offering valuable insights into the intricate dynamics of motorsport. By leveraging Formula 1 data from 2014 to 2023 and employing advanced machine learning techniques, specifically LSTMs, this research has successfully highlighted both the challenges and opportunities in applying predictive analytics to Formula 1 racing. The study began with a comparative analysis of various machine learning models which allowed us to later establish baseline performances with simpler models such as linear regression, decision trees, and random forests. These models provided a starting point for understanding the complexities of the dataset and served as benchmarks for evaluating more sophisticated approaches. The LSTM models, designed specifically to handle sequential data, demonstrated superior performance in capturing temporal dependencies and relative driver performance within races. However, the results also revealed that minimizing test loss alone does not always translate to better race outcome predictions. This insight underscores the importance of incorporating additional evaluation metrics, such as position loss and historical loss, to align model objectives with real-world racing dynamics. The results from testing on the Abu Dhabi Grand Prix and Bahrain Grand Prix further validated the effectiveness of the LSTM models. While earlier versions of the LSTM models struggled with outlier predictions and failed to capture dominant performances accurately, refinements in model architecture and training methodology—such as adding dropout layers, increasing epochs, and introducing custom loss functions—led to significant improvements. The later iterations (v2.1, v2.2, v2.3, and v2.4) demonstrated remarkable accuracy in predicting top positions and race outcomes for the 2024 Bahrain Grand Prix and 2024 Abu Dhabi Grand Prix, showcasing the potential of these models when given the right data. For future work, further research can include looking into the same subject scope through the primary lens of transformers instead of LSTMs. This may help by leveraging transformer models' ability to capture global dependencies for improved lap time predictions. Then, one can see whether the transformer models outperform the LSTM models we have seen so far. Moreover, future research can also try to increase the prediction scope. Right now, only lap times are being predicted, but for a model that can 'truly' predict lap times for a given race, one must also be able to predict pitstop strategy (pitting lap and tire compound) as well as safety car presence in a lap. Predicting these would result in a more complete predictive model that can predict the outcomes of a race before it has happened.

REFERENCES

[1] FORMULA 1. 2023. Beginner's Guide to F1. https://www.youtube.com/watch?v=Q-jjZMMxbZs

[2] Formula 1. 2024. F1 - The Official Home of Formula 1® Racing. https://www.formula1.com/en/results/2024/races/1252/abu-dhabi/fastest-laps

[3] Formula 1. 2024. Relive the action from the season finale in Abu Dhabi. https://www.formula1.com/en/latest/article/highlights-relive-the-action-from-the-season-finale-in-abu-dhabi-as-norris.6hiM6GURCnP0FRiaMfCVk3

[4] Chain Bear. 2017. Basics of F1 Race Strategy. https://www.youtube.com/watch?v=wqf-dJyU_WA

[5] Adam Cooper. 2022. Why Latifi is struggling with the F1 2022 Williams. https://www.autosport.com/f1/news/why-latifi-is-struggling-with-the-f1-2022-williams/10247263/

[6] Carla De Francesco, Luigi De Giovanni, Marco Ferrante, Giovanni Fonseca, Francesco Lisi, and Silvia Pontarollo. 2017. *Proceedings of MathSport International 2017 Conference*. Padova University Press. 87–96 pages. https://www.padovauniversitypress.it/en/publications/9788869380587

[7] Data Headhunters. 2024. Decision Trees vs Random Forests: Comparing Predictive Power. https://dataheadhunters.com/academy/decision-trees-vs-random-forests-comparing-predictive-power/

[8] García Loreto and Tejada. 2023. *Applying Machine Learning to Forecast Formula 1 Race Outcomes*. https://aaltodoc.aalto.fi/server/api/core/bitstreams/70d5a580-c282-4278-8462-94d061471546/content

[9] Veronica Nigro. 2020. Formula 1 Race Predictor. https://towardsdatascience.com/formula-1-race-predictor-5d4bfae887da

[10] D Politecnico and Milano. 2020. *Open Loop Planning for Formula 1 Race Strategy identification*. https://www.politesi.polimi.it/bitstream/10589/175624/3/2021_04_Piccinotti.pdf

[11] Priya Shelke, Anurag Pande, Srujan Kale, Yash Paralikar, and Atul Kulkarni. 2023. F1 Race Winner Predictor. In *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. 1–4. https://doi.org/10.1109/ICCUBEA58933.2023.10392224

[12] Léon Sobrie. 2020. *SIFTING THROUGH THE NOISE IN FORMULA ONE: PREDICTIVE PERFORMANCE OF TREE-BASED MODELS*. https://libstore.ugent.be/fulltxt/RUG01/002/837/806/RUG01-002837806_2020_0001_AC.pdf

[13] William Villegas-Ch, Joselin García-Ortiz, and Angel Jaramillo-Alcazar. 2023. An Approach Based on Recurrent Neural Networks and Interactive Visualization to Improve Explainability in AI Systems. *Big Data and Cognitive Computing* 7, 3 (2023). https://doi.org/10.3390/bdcc7030136

[14] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2023. *Transformers in Time Series: A Survey*. https://www.ijcai.org/proceedings/2023/0759.pdf

[15] Intan Dea Yutami. 2022. Formula 1 Races Analysis. https://intandeay.medium.com/f1-f11dcd91d025

A APPENDIX: ADDITIONAL FIGURES

	raceId	circuitId	driverId	constructorId	grid	year	round	lap	milliseconds	q1milli	q2milli	q3milli	Driver_Seasc	driverwins	YOB	Races_befor	Races_won	Podiums	isSafetyCar	isSafetyCarPrev
0	900	1	1	131	1	2014	1	1	106128	91.699	102.89	104.231	0	0	1985	129	22	53	0	0
1	900	1	1	131	1	2014	1	2	100287	91.699	102.89	104.231	0	0	1985	129	22	53	0	0
2	900	1	3	131	3	2014	1	1	102038	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
3	900	1	3	131	3	2014	1	2	97687	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
4	900	1	3	131	3	2014	1	3	95765	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
5	900	1	3	131	3	2014	1	4	94939	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
6	900	1	3	131	3	2014	1	5	95438	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
7	900	1	3	131	3	2014	1	6	94977	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
8	900	1	3	131	3	2014	1	7	95417	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
9	900	1	3	131	3	2014	1	8	94550	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
10	900	1	3	131	3	2014	1	9	94217	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
11	900	1	3	131	3	2014	1	10	94364	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
12	900	1	3	131	3	2014	1	11	95185	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
13	900	1	3	131	3	2014	1	12	134269	92.564	102.264	104.595	0	1	1985	132	3	15	1	0
14	900	1	3	131	3	2014	1	13	149953	92.564	102.264	104.595	0	1	1985	132	3	15	1	1
15	900	1	3	131	3	2014	1	14	141208	92.564	102.264	104.595	0	1	1985	132	3	15	1	1
16	900	1	3	131	3	2014	1	15	134486	92.564	102.264	104.595	0	1	1985	132	3	15	1	1
17	900	1	3	131	3	2014	1	16	93976	92.564	102.264	104.595	0	1	1985	132	3	15	1	1
18	900	1	3	131	3	2014	1	17	94040	92.564	102.264	104.595	0	1	1985	132	3	15	0	1
19	900	1	3	131	3	2014	1	18	93195	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
20	900	1	3	131	3	2014	1	19	92478	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
21	900	1	3	131	3	2014	1	20	93331	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
22	900	1	3	131	3	2014	1	21	92839	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
23	900	1	3	131	3	2014	1	22	93144	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
24	900	1	3	131	3	2014	1	23	93213	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
25	900	1	3	131	3	2014	1	24	93936	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
26	900	1	3	131	3	2014	1	25	93941	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
27	900	1	3	131	3	2014	1	26	94539	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
28	900	1	3	131	3	2014	1	27	95588	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
29	900	1	3	131	3	2014	1	28	95416	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
30	900	1	3	131	3	2014	1	29	94069	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
31	900	1	3	131	3	2014	1	30	94946	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
32	900	1	3	131	3	2014	1	31	94272	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
33	900	1	3	131	3	2014	1	32	95305	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
34	900	1	3	131	3	2014	1	33	94088	92.564	102.264	104.595	0	1	1985	132	3	15	0	0
35	900	1	3	131	3	2014	1	34	93979	92.564	102.264	104.595	0	1	1985	132	3	15	0	0

Driver_Season_Points	driverwins	YOB	Races_before	Races_won	Podiums	isSafetyCar	isSafetyCarPrev	isPitting	tyre_age	tyre_compou	isVET	isZHO	isVER	isTSU	isSTR	isMSC	isSAR	isRIC
0	0	1985	129	22	53	0	0	0	1	-1	0	0	0	0	0	0	0	0
0	0	1985	129	22	53	0	0	0	2	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	1	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	2	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	3	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	4	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	5	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	6	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	7	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	8	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	9	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	10	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	11	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	1	0	1	12	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	1	1	0	1	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	1	1	0	2	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	1	1	0	3	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	1	1	0	4	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	1	0	5	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	6	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	7	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	8	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	9	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	10	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	11	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	12	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	13	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	14	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	15	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	16	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	17	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	18	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	19	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	20	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	21	-1	0	0	0	0	0	0	0	0
0	1	1985	132	3	15	0	0	0	22	-1	0	0	0	0	0	0	0	0

Figure 1: Snapshot of the Dataset

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
hamilton	10.392419	10.154527	9.753321	9.527362	9.405726	9.338777	9.30169	9.28113	9.269845	9.263825	9.260807	9.259494	9.259137	9.259295	10.223306	10.859659	9.90342
alonso	9.910796	10.061363	9.67752	9.4635725	9.349444	9.285577	9.2492285	9.228513	9.216766	9.210158	9.206464	10.00323	10.748145	9.898217	9.581088	9.409502	9.311471
hulkenberg	10.264563	10.27868	9.830133	9.598249	9.475173	9.405444	9.364628	9.340119	9.324916	9.315064	9.308332	9.303477	10.037667	11.020971	10.056638	9.713204	9.5258465
perez	9.900034	9.955861	9.602288	9.411341	9.30403	9.2399025	9.200524	9.175637	9.159243	9.147849	9.139434	9.132824	9.127332	9.122543	9.118203	9.1141405	9.925471
ricciardo	10.393998	10.221999	9.80014	9.582739	9.466246	9.400266	10.134547	10.914937	10.06331	9.7332325	9.552446	9.448869	9.3895235	9.355864	9.337211	9.327393	9.322827
bottas	10.325883	10.094572	9.746552	9.561308	9.46222	9.4058485	9.372709	9.35271	9.340224	9.332044	9.32632	9.321979	9.318395	9.315195	9.31216	9.30915	9.306078
kevin_magnu	10.553107	10.263282	9.860194	9.652887	10.29508	10.880677	10.109759	9.797635	9.6235895	9.521514	9.460303	9.422463	9.397975	9.381108	9.368608	9.358644	9.3502035
max_verstap	9.94284	9.945194	9.409912	9.12721	8.978624	8.898174	8.855175	8.833352	8.823504	8.820321	8.8207035	8.822829	8.8256235	8.828446	8.830918	9.763958	11.068215
sainz	10.134467	10.057729	9.685452	9.490819	9.386884	9.32606	9.288066	9.262734	9.244481	9.230163	9.21801	9.207044	9.19674	9.186824	9.177171	9.167726	9.158469
ocon	10.360775	10.225587	9.823066	9.61738	9.50721	9.443447	9.404929	9.380776	9.3649435	9.354009	9.346031	9.33992	9.335066	9.331128	10.079632	10.8686	9.986222
stroll	10.336966	10.223544	9.815226	9.607026	9.496308	9.432479	9.393927	9.369617	9.353435	9.341918	9.333105	9.325907	9.319724	9.314241	9.309293	9.304795	9.300706
gasly	10.247558	10.241831	9.850135	9.646085	9.538001	9.476396	9.439813	9.417348	9.402985	9.393334	9.386481	9.381351	9.377348	9.374141	9.371541	9.369439	10.139894
leclerc	9.936349	9.89965	9.552265	9.371842	9.272202	9.213248	9.177507	9.155417	9.141412	9.132236	9.125989	9.12156	9.118283	9.115754	9.113713	9.111983	9.94999
norris	9.985477	9.982241	9.648193	9.462797	9.359029	9.297904	9.260373	9.236155	9.219514	9.207232	9.1975	9.1893015	9.182074	9.991611	10.660956	9.838128	9.5582
russell	9.940876	10.003676	9.638799	9.439382	9.331158	9.26884	9.231402	9.208011	9.192708	9.182151	9.174447	9.168528	9.1637945	9.932925	10.687099	9.828942	9.535867
albon	10.371836	10.157043	9.78452	9.595321	9.494243	9.435195	9.398719	9.374948	9.358431	9.346095	9.336209	9.327816	9.320392	9.313659	9.307475	10.063058	10.736375
tsunoda	10.196336	10.146868	9.743388	9.535442	9.424809	9.361369	9.323425	9.299907	9.284697	9.274346	9.26691	9.261299	9.256913	9.253418	9.250632	9.248445	9.24679
zhou	10.606307	10.138489	9.788416	9.613305	9.519411	9.463636	9.428103	9.403802	9.385753	9.371143	9.358382	9.346593	10.088069	10.637058	9.915296	9.663701	9.5253
piastri	10.223572	10.079272	9.730218	9.543774	9.443177	9.385077	9.349621	9.326708	9.310854	9.299	9.289436	9.281216	10.109998	10.703784	9.903937	9.634358	9.486821
sargeant	9.28239	8.771414	8.669437	8.602786	8.557865	8.524151	8.496557	8.472347	8.450013	8.428738	8.408106	8.3879385	8.368192	8.348906	8.7038355	8.889254	8.64858

Figure 2: Snapshot of the Predictions from the 2023 Abu Dhabi Grand Prix

Research Article

Erik-Jan van Kesteren* and Tom Bergkamp

Bayesian analysis of Formula One race results: disentangling driver skill and constructor advantage

https://doi.org/10.1515/jqas-2022-0021
Received March 15, 2022; accepted June 2, 2023;
published online July 25, 2023

Abstract: Successful performance in Formula One is determined by combination of both the driver's skill and race-car constructor advantage. This makes key performance questions in the sport difficult to answer. For example, who is the best Formula One driver, which is the best constructor, and what is their relative contribution to success? In this paper, we answer these questions based on data from the hybrid era in Formula One (2014–2021 seasons). We present a novel Bayesian multilevel rank-ordered logit regression method to model individual race finishing positions. We show that our modelling approach describes our data well, which allows for precise inferences about driver skill and constructor advantage. We conclude that Hamilton and Verstappen are the best drivers in the hybrid era, the top-three teams (Mercedes, Ferrari, and Red Bull) clearly outperform other constructors, and approximately 88 % of the variance in race results is explained by the constructor. We argue that this modelling approach may prove useful for sports beyond Formula One, as it creates performance ratings for independent components contributing to success.

Keywords: multilevel model; racing; ranking; sports performance

1 Introduction

In most competitive sports with a large individual component (e.g., chess, athletics, swimming, or tennis) success is determined primarily by the relative skill (i.e., ability or talent) of the contestants. Official competitions in such sports naturally result in official rankings representing this skill level on an individual basis. Unlike these skill-based sports, competitive motor racing has an additional key factor contributing to success: the material, i.e., the race car or motor bike. In Formula One in particular, the influence of the car on the results is considered to be substantial (Budzinski and Feddersen 2020). In contrast to "spec" series, where cars have the same specifications and are built by the same constructor, Formula One cars are each built from the ground up by different constructors with differing levels of technological and financial resources. These resource gaps can lead to large differences in performance between cars, despite rules imposed to counteract such performance differences. Arguably, the presence of these large differences in constructor advantage have led to a single constructor (Mercedes) dominating the sport in most of the "hybrid era", from 2014 to 2020.

The dependence on materials reduces the relative influence of driver skill on success — a race win is an entangled combination of both driver and constructor performance. Therefore, ranking drivers in terms of skill level by simply using competition race results is complex. Because of this problem, perennial questions such as "who is the best Formula One driver?", "which constructor is the best?" and "is the driver or the constructor-team more important to success?" are difficult to answer scientifically. These questions are persistent in the sport; the 2016 world champion Niko Rosberg famously posed that 80 % of success in Formula One can be attributed to the car and 20 % to the driver (Bol 2020). In this article, we argue that it is now possible to answer these questions due the widespread availability of race result data and the accessibility of advanced statistical methodology. We propose a novel Bayesian multilevel regression model to answer three interrelated questions for the hybrid era in Formula One: (a) what is the relative influence of the driver and constructor on race results, (b) how do drivers rank in terms of skill level, and (c) how do constructors rank in terms of race car advantage?

***Corresponding author: Erik-Jan van Kesteren**, Methodology & Statistics, Utrecht University, Utrecht, Netherlands,
E-mail: e.vankesteren1@uu.nl. https://orcid.org/0000-0003-1548-1663
Tom Bergkamp, Royal Dutch Football Association, Zeist, Netherlands,
E-mail: tom.bergkamp@knvb.nl

Several attempts have been made to disentangle driver and constructor performance. Eichenberger and Stadelmann (2009) used linear regression with dummy variables and several covariates to estimate driver and constructor-year effects on race finishing position in the 1950-to-2006 Formula One seasons. The driver-specific effects on race outcomes were then used to compute a ranking of drivers' skill level. Additionally, the authors found that predictors for weather (wet vs. dry) and circuit type (street circuit vs. permanent circuit) were relevant additions to the model. One shortcoming of this study is the choice of outcome variable: because the number of contestants per race changed over seasons (and sometimes even within seasons), the interpretation of "finishing position" changed as well. This shortcoming was addressed by Phillips (2014), who analysed Formula One race data from 1950-to-2013. Here, the effects of driver performance, constructor performance and season difficulty were estimated using an adjusted "points scored" outcome variable, based on the official points scoring system used in the Formula One between 1991 and 2002: 10 points for first place, 6 for second, down to 1 point for sixth place. With this approach, the authors ranked drivers in terms of skill and concluded that Juan Manuel Fangio was the best driver of all time.

As Phillips (2014) noted, there are two reasons that such models can differentiate driver from constructor effects: (a) throughout the history of Formula One, constructors have generally had two cars enter a race. Barring minor differences in individual races, these cars have the same performance, which allows for direct comparisons of a driver's skill level against their teammates. Furthermore, (b) drivers generally move to different constructor-teams throughout their career, meaning their teammates also change. This allows for simultaneous estimation of driver and constructor effects based on race results.

A disadvantage of the aforementioned studies is that they used models with many dummy variables (fixed effects) instead a multilevel (random effects) model. A multilevel approach is beneficial, as it makes the models tractable with fewer observations per driver and improves predictions for nested data (Gelman 2006). A similar argument was made by Bell et al. (2016), who used the same "points scored" outcome variable as Phillips (2014), but used a multilevel (random-coefficients) linear model to determine the driver and constructor-year effects. Using this approach, the authors were the first to directly estimate a parameter for comparing driver skill and constructor advantage: they concluded that the constructor accounts for 86 % of the variance in points scored, and the driver for 14 %.

While the approach presented by Bell et al. (2016) provides an answer to the research questions posed above (pre-2014), we here present several improvements which enable a more accurate insight into the current state of Formula One. First and foremost, we argue that the "points scored" variable leads to information loss: any result below sixth place leads to zero points, making it impossible to differentiate constructors and drivers who consistently finish below this threshold. Our approach, explained in Section 3, is to create a Rank-Ordered Logit model for the rank ordering of finish positions for each race. Second, we focus on the current hybrid era in Formula One, from 2014 to 2021, using data from a publicly available resource (Newell 2021). This focus enables us to closely inspect changes across these most recent seasons with comparable regulations. Third, we apply a Bayesian workflow for model development (Gelman et al. 2013), visualisation (Gabry et al. 2019), and comparison (Vehtari, Gelman, and Gabry 2017), which makes the parameters interpretable and clarifies the model's implications. Using our approach, the parameters for driver skill and constructor advantage in our model are directly interpretable as log-odds ratios of beating competitors, similar to Elo ratings in chess (e.g., Van Der Maas and Wagenmakers 2005). These parameters (and their uncertainty intervals) can then be used to create rankings which provide clear insight into the relative performance of drivers and constructors in Formula One.

The paper is structured as follows. First, in Section 2 we describe the data source and processing steps we performed to obtain the predictors and the outcome of interest. Then, in Section 3 we introduce the proposed Bayesian multilevel regression model and its interpretation. In the same section, we also perform model selection and model checking to validate the estimation procedure. In Section 4 we perform inference for the 2014-to-2021 Formula One seasons to answer the research questions surrounding driver skill and constructor advantage. This includes a driver ranking for the 2021 season and investigation of a counterfactual statement based on the estimated model: would Hamilton in an Alfa Romeo beat Räikkönen in a Mercedes? Last, in Section 5 we place our contributions in context, discuss its implications, and provide suggestions for future work. All analysis scripts and pre-processed data are openly available in the Supplementary material at https://doi.org/10.5281/zenodo.7632045 (van Kesteren and Bergkamp 2023).

2 Data processing

We collected race results (driver id, constructor id, season (year), race number, finishing position and status) for the 2014-to-2021 Formula One seasons from the dataset behind the Ergast motorsports API (Newell 2021). This resulted in a dataset of 160 races, with 51 unique drivers and 19 distinct constructors. In addition, we performed a data enrichment step by scraping and parsing further race information from Wikipedia. In this step, two predictors were added to the dataset which were previously found to be relevant in the work by Eichenberger and Stadelmann (2009) and Bell et al. (2016). We constructed a variable indicating whether the race was wet or dry, and we also collected information about the circuit type (street circuit or permanent circuit). As the information was not complete, especially for weather type in the 2018 season, we manually completed this data using publicly available race summaries. In total, in the period of interest, there were 143 dry races and 17 wet races.

For the main analysis, non-finishers were removed from the analysis, removing 590 rows (from the original 3267 rows) in the dataset. Thus, we only examined finished races for each driver: accidents and other reasons for non-finishing or non-starting can be due to any number of reasons, which adds noise and complexity to the outcome of interest. This removal of non-finishes has implications for the interpretation of the model, which we discuss in Section 3. In Appendix A we perform a sensitivity analysis for the effects of dealing with finishing in different ways. Note that instead of the approach we take, it is also possible to include non-finishing in the analysis by creating a larger, joint model of non-finishes and race results conditional on finishing (e.g., Ingram 2021).

A visual display of the finish positions for three drivers (Räikkönen, Hamilton, and Giovinazzi) in the 2015–2020 seasons is shown in Figure 1. This forms the basis of the outcome variable we construct, which is the per-race ranking of competitors. By considering only race ranking as our outcome, we disregard the starting position of the competitor, which is determined in a qualifying session before the race. Thus, any conclusions about relative performance of competitors necessarily includes both race performance

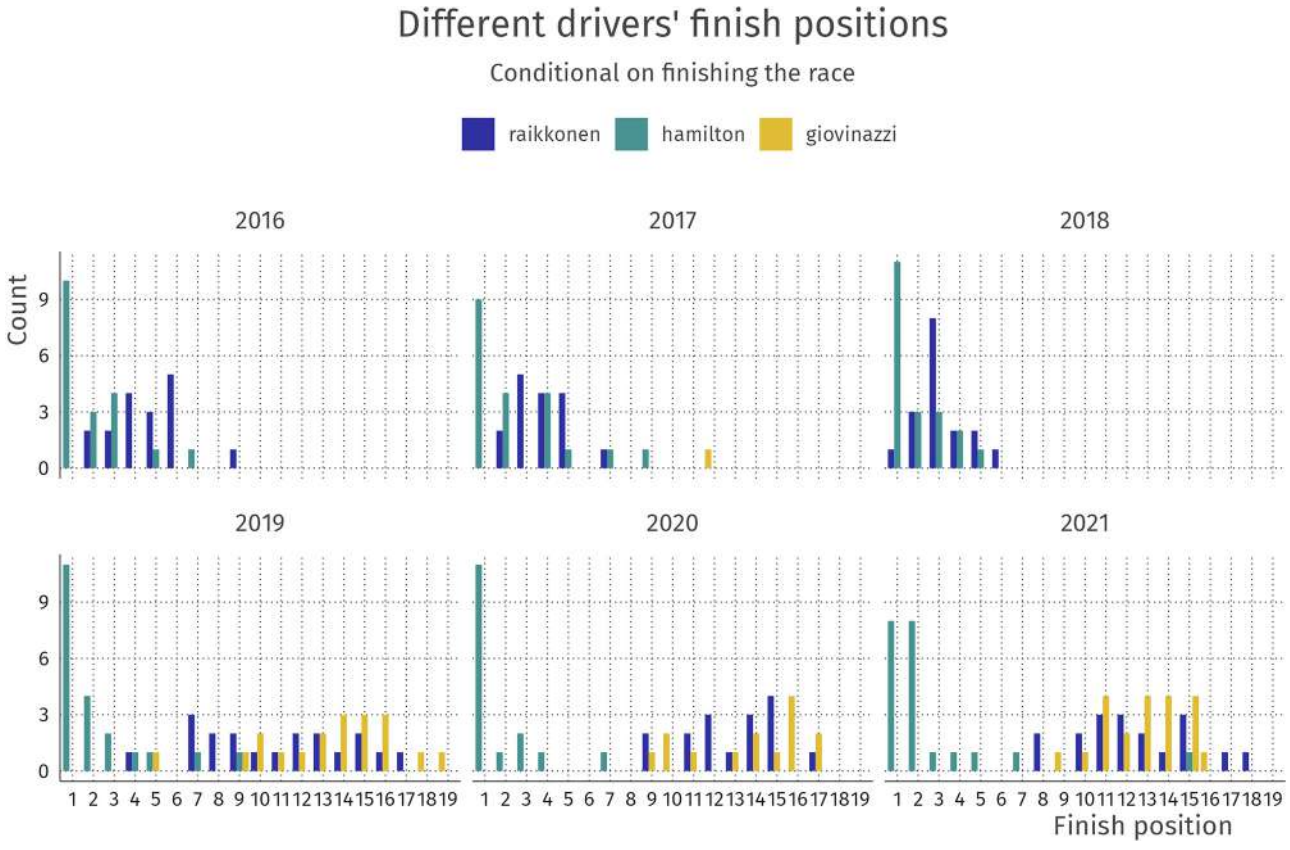


Figure 1: Finish positions for Räikkönen, Hamilton, and Giovinazzi for the seasons 2015–2020. Räikkönen's move from Ferrari to Alfa Romeo in 2019 is clearly visible in the finish positions. At Alfa Romeo, Räikkönen became teammates with Giovinazzi, who he on average outperforms slightly.

and qualifying performance: for all of our models, a race-winning performance is no different if the competitor starts the race in first or last position. In the next section, we explain in detail how we model this outcome variable.

3 Bayesian multilevel rank-ordered logit model

In this section, we describe in detail the process used to model the finish position as a function of driver skill, yearly driver form, constructor advantage, and yearly constructor form. All models in this paper were estimated using the software package `Stan` (Carpenter et al. 2017) with reasonable default priors for all parameter types (see Section 2.1 of Bürkner (2017) for more details).

3.1 Basic model specification and parameter interpretation

For each race r, assume we have a set of competitors C_r which compete for the win. The outcome variable in our model is a vector representing a ranking of these competitors, $\boldsymbol{y}_r$. In our generative model, we assume that this ranking follows a Rank-Ordered Logit (ROL) distribution, based on a vector of latent abilities of the competitors in that race $\boldsymbol{\vartheta}_r$:

$$\boldsymbol{y}_r \sim \text{RankOrderedLogit}(\boldsymbol{\vartheta}_r) \tag{1}$$

The precise definition of the ROL distribution is explained in detail in Glickman and Hennessy (2015), and in our extended model definition in Appendix B. Here, note that each element of $\boldsymbol{\vartheta}_r$ represents the latent ability of each competitor in the race:

$$\boldsymbol{\vartheta}_r = \{\vartheta_c \mid c \in C_r\} \tag{2}$$

A competitor $c \in C_r$ is defined as a pairing of a driver d and constructor (or team) t, in a particular season (or year) s. Thus, we represent the skill of each competitor as the skill of the driver-constructor pairing in a season: $\vartheta_c = \theta_{dts}$. In our model, this skill is a sum of the average driver skill θ_d, the driver's seasonal form θ_{ds}, the constructor's average advantage θ_t and the constructor's seasonal form θ_{ts}. This results in the following cross-classified multilevel model for each competitor's latent ability:

$$\begin{aligned}
\vartheta_c = \theta_{dts} &= \theta_d + \theta_{ds} + \theta_t + \theta_{ts} \\
\theta_d &\sim \mathcal{N}\left(0, \sigma_d^2\right) \\
\theta_{ds} &\sim \mathcal{N}\left(0, \sigma_{ds}^2\right) \\
\theta_t &\sim \mathcal{N}\left(0, \sigma_t^2\right) \\
\theta_{ts} &\sim \mathcal{N}\left(0, \sigma_{ts}^2\right)
\end{aligned} \tag{3}$$

This model form leads to a specific, natural interpretation for the parameters θ. The logit link function in the ROL likelihood, in combination with the omission of an overall intercept, ensures that the (hypothetical) average driver at an average constructor with an average seasonal form will on average have $\theta_{dts} = 0$, which translates into a probability of 0.5 of beating a randomly selected other competitor. Then, θ_d represents the mean driver skill as a log-odds ratio; e.g., if $\theta_d = 0.3$, this means (ceteris paribus) that the probability of finishing in front of the other driver is $1/(1 + \mathrm{e}^{-0.3}) \approx 0.57$. This parameter represents a deviation from the average driver, so negative values mean worse than average skill, and positive values mean better than average skill. We also include the seasonal driver form parameter θ_{ds}, which represents yearly deviations from this long-term average driver skill.

A similar interpretation holds for θ_t, which indicates the long-term average constructor advantage. Constructors with positive values on this parameter tend to produce cars which are better than average, and negative values indicate cars which are worse on average. We also include the seasonal constructor form parameter θ_{ts}, which represents yearly deviations from this long-term average team advantage. In Appendix D we compare this parameterization to two other ways of dealing with time-varying θ parameters. For more detailed parameter interpretations and conclusions, see Section 4.

Note that the latent skill are assumed to be stable within seasons. This means that for races within the same season s with the same competitors C_r we assume independent and identically distributed rankings $\boldsymbol{y}_r$. Note also that with this model formulation we implicitly assume no correlation between the random intercepts for driver and car constructor; there are no interactions at all between driver skill and team advantage. This means that a driver's skill is independent of the team advantage, i.e., the driver skill does not change when the driver moves to a different constructor.

3.2 Extending the basic model

Previous work has shown that several predictors may change the race results (Bell et al. 2016). The first extension we make reflects the knowledge that wet races are different from dry races. Wet races require a specific set of skills, which rely less on the car and more on the driver. Like Bell et al. (2016), we represent this knowledge by splitting the driver average skill parameter into a random intercept

parameter γ_{0d} and a random slope parameter γ_{1d} as in Equation (4):

$$\theta_d = \gamma_{0d} + \gamma_{1d} \cdot \text{wet\_race} \tag{4}$$

where wet_race is an indicator (dummy) variable with a 1 if the race was wet and 0 if the race was dry (see Section 2 for details). The driver average skill in dry races is then γ_{0d}, and in wet races it is $\gamma_{0d} + \gamma_{1d}$.

The second extension we make reflects the knowledge that different constructors have different car philosophies. Theoretically, high-downforce concept cars (e.g., Red Bull cars in the hybrid era) are relatively better suited to narrow, curvy street circuits such as the famous Monaco circuit. However, this advantage disappears on fast, permanent circuits such as Monza (Italy). Therefore, we add a random slope to the constructor advantage parameter, splitting it up as in Equation (5):

$$\theta_t = \gamma_{0t} + \gamma_{1t} \cdot \text{permanent\_circuit} \tag{5}$$

where permanent_circuit is an indicator (dummy) variable with a 1 if the race was on a permanent circuit and 0 if the race was on a street circuit (see Section 2 for details).

With these two extensions, four models are possible: (a) the basic model, (b) a weather model, (c) a circuit type model, and (d) a weather and circuit type model. In the next subsection, we compare these models to select our final model, on which we perform inference.

3.3 Model selection

We used efficient leave-one-out cross-validation (LOO, Vehtari, Gelman, and Gabry 2017) to compare the four possible models. In short, LOO uses the samples from the posterior to compute the expected log posterior density (ELPD) for each model, which is an alternative to the standard information criteria in Bayesian model comparison such as the Bayes Factor (marginal density) or DIC. For the four tested models, the results are shown in Table 1.

Table 1: Model comparison results showing the expected log posterior density (ELPD), its standard error, the difference between each model and the best model, and the standard error of this difference. Note: a higher ELPD indicates a better fit.

	ELPD	SE_{ELPD}	Δ	SE_{Δ}
Circuit	−3992.29	84.11		
Basic	−3993.36	83.72	−1.07	6.35
Circuit + weather	−3993.66	84.31	−1.37	1.94
Weather	−3995.55	84.09	−3.25	6.22

The circuit model is shown to be the best model in terms of ELPD, but the differences are small compared to the standard errors of these differences. This indicates that in terms of out-of-sample predictive performance, the models perform similarly. To maintain parsimony in both modelling and interpretation, we decide use the basic model for assessment, inference, and prediction in the following sections.

3.4 Model assessment

Posterior samples were obtained via Hamiltonian Monte Carlo sampling with 8 chains of 1250 samples each after 1000 burn-in iterations. The effective sample size for all parameters was higher than 2500, and their R-hat value was smaller than 1.01, indicating adequate convergence. Trace plots for the main parameters of the model are shown in Appendix C.

Posterior predictive checks (PPCs) are a vital part of the Bayesian workflow (Gabry et al. 2019). In a visual PPC, simulated data $\tilde{y}$ from the posterior predictive distribution is compared to the observed data y. If $\tilde{y}$ approximates y well, then the model captures the outcome well. For our model, we performed two PPCs: one for the 2015 season (early hybrid era) and one for the 2019 season (late hybrid era). The results are shown in Figures 2 and 3.

The plots show satisfactory recovery of individual performances per driver-constructor-season combination. It is noteworthy that for some drivers, the observed data show bimodality (e.g., Vettel in 2019 or Bottas in 2015) while the model cannot capture these patterns. In these cases, the averages in the observed and simulated data still seem to align closely. In general, we conclude that the model fits the observed data well. In the next section, we use the model to perform parameter inference.

4 Results

In this section, we perform inference with the model that resulted from the modelling procedure described in Section 3. To narrow down our inference efforts, we focus on a subset of the drivers and teams competing in the 2021 season. The results for all drivers, teams, and seasons are available in the Supplementary material (van Kesteren and Bergkamp 2023).

4.1 Driver skill

After estimation of the preferred model, it is possible answer the question of which driver is the most skilled,

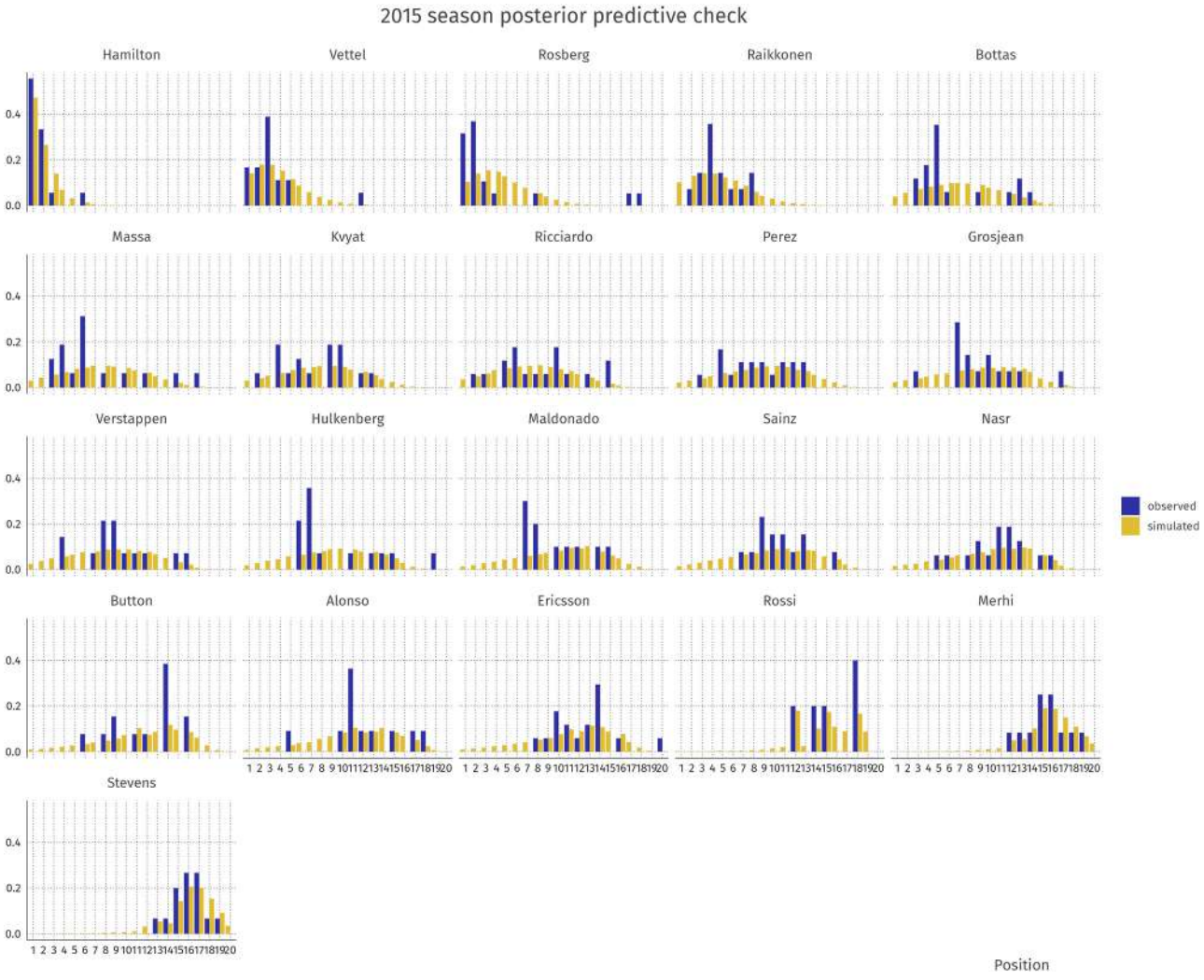


Figure 2: Posterior predictive check for driver finishing positions in the 2015 season.

while taking into account constructor advantage, constructor form, as well as all parameter uncertainties. In order to produce a ranking, we obtained the posterior means and 89 % credible intervals (see McElreath 2018) of $\theta_d + \theta_{ds}$ for the season 2021. These summaries are shown in Figure 4.

In Figure 4 it is apparent that of the 2021 drivers, Hamilton and Verstappen are ranked as the most skilled driver. Note that this ranking comes directly from the model, which is estimated only on the hybrid-era data. Earlier performances by drivers such as Vettel (four-time world champion in the period 2010–2013) and Räikkönen (world champion in 2007) have not been taken into account, explaining their lower position on the ranking.

Because the model contains a yearly form parameter, we can also visualize the latent skill trajectories of several drivers throughout the hybrid era, with their credible intervals. The result of this visualization for 12 drivers of the 2021 season is shown in Figure 5.

The figure shows that there are slight changes in skill across seasons. Drivers such as Verstappen, Norris, and Gasly tend to improve over years, on average, whereas Bottas displays a slight decline. Notably, Hamilton is consistently at the top, whereas Alonso and Sainz are consistently slightly above average in terms of their latent skill.

4.2 Constructor advantage

For the constructors competing in 2021, we here investigate how much of an advantage their car yields. We do this by computing the posterior means and 89 % credible intervals of θ_t from the model. The result is shown in Figure 6.

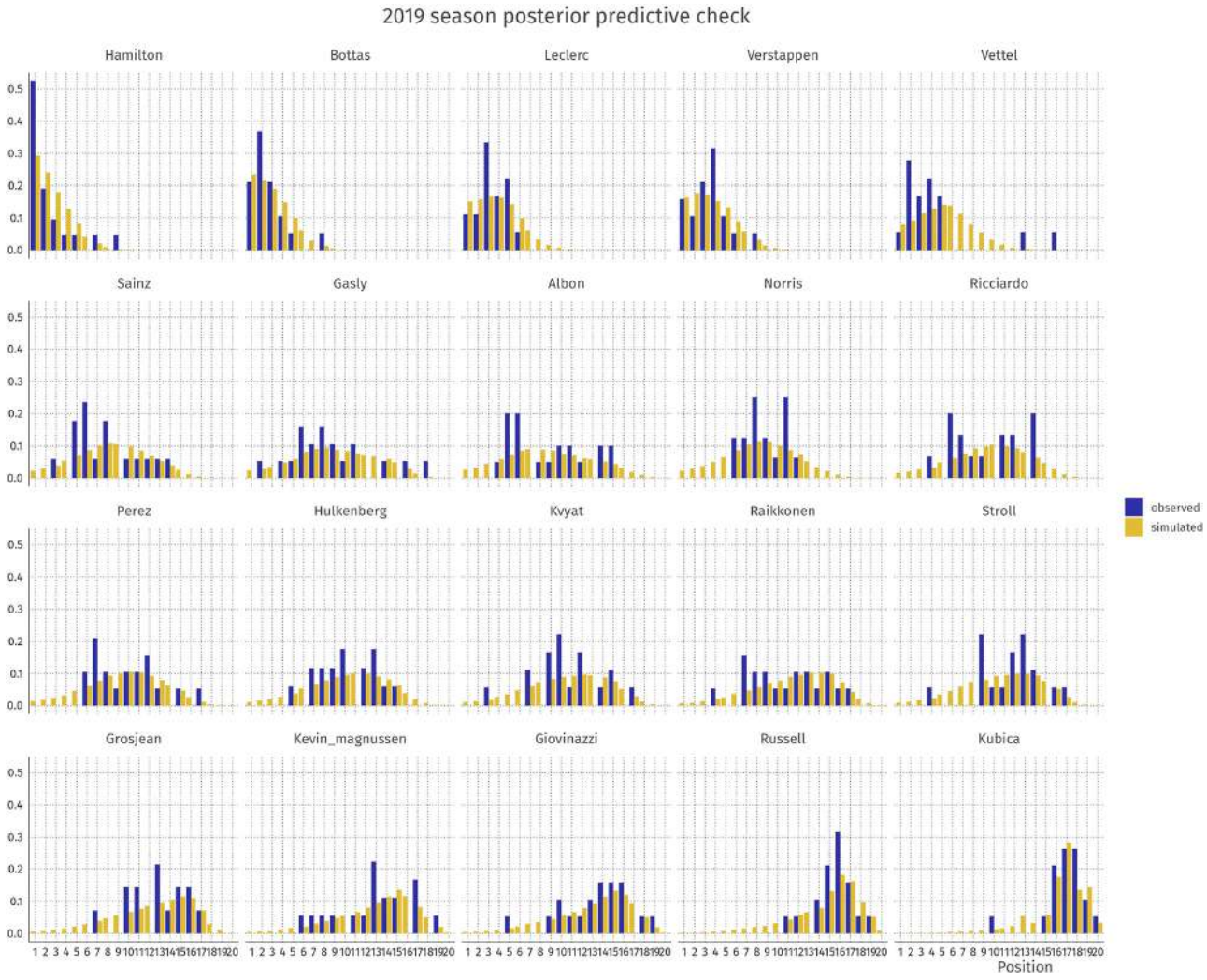


Figure 3: Posterior predictive check for driver finishing positions in the 2019 season.

One feature that becomes clear from this constructor advantage plot is the relative advantage of the *big three* teams: Mercedes, Ferrari, and Red Bull have the largest budget and the most resources to spend on developing their car, which has resulted in these three teams excelling in the hybrid era. Another interesting feature in this plot is the large uncertainty around the teams that have competed in only a few seasons. For example, Alpine and Aston Martin were new teams in 2021, and therefore have not had a chance compete in many races, resulting in uncertainty around where it is placed in this constructor ranking. Again, these parameters need to be interpreted with care: they represent the average constructor advantage over the entire hybrid era.

The last random intercept component is the constructor-year effects θ_{ts}. These represent yearly constructor form, as a deviation from their long-term average advantage. In Figure 7, the yearly constructor advantage trajectories for a selection of teams is shown from 2014 to 2021.

One of the most striking features from this graph is Ferrari's drop in form from 2019 to 2020. There is a good explanation for this: after the 2019 season, there were allegations that Ferrari's engine in the previous year did not match league regulations. Ferrari subsequently reached a settlement with the Formula One Management (Formula1.com 2020), which left the team with a relatively weak engine in the 2020 season. This change is aptly reflected in the constructor form parameter for Ferrari, but also to a lesser extent in that of McLaren, which profited from Ferrari's drop and ended up third in the constructor's championship that year.

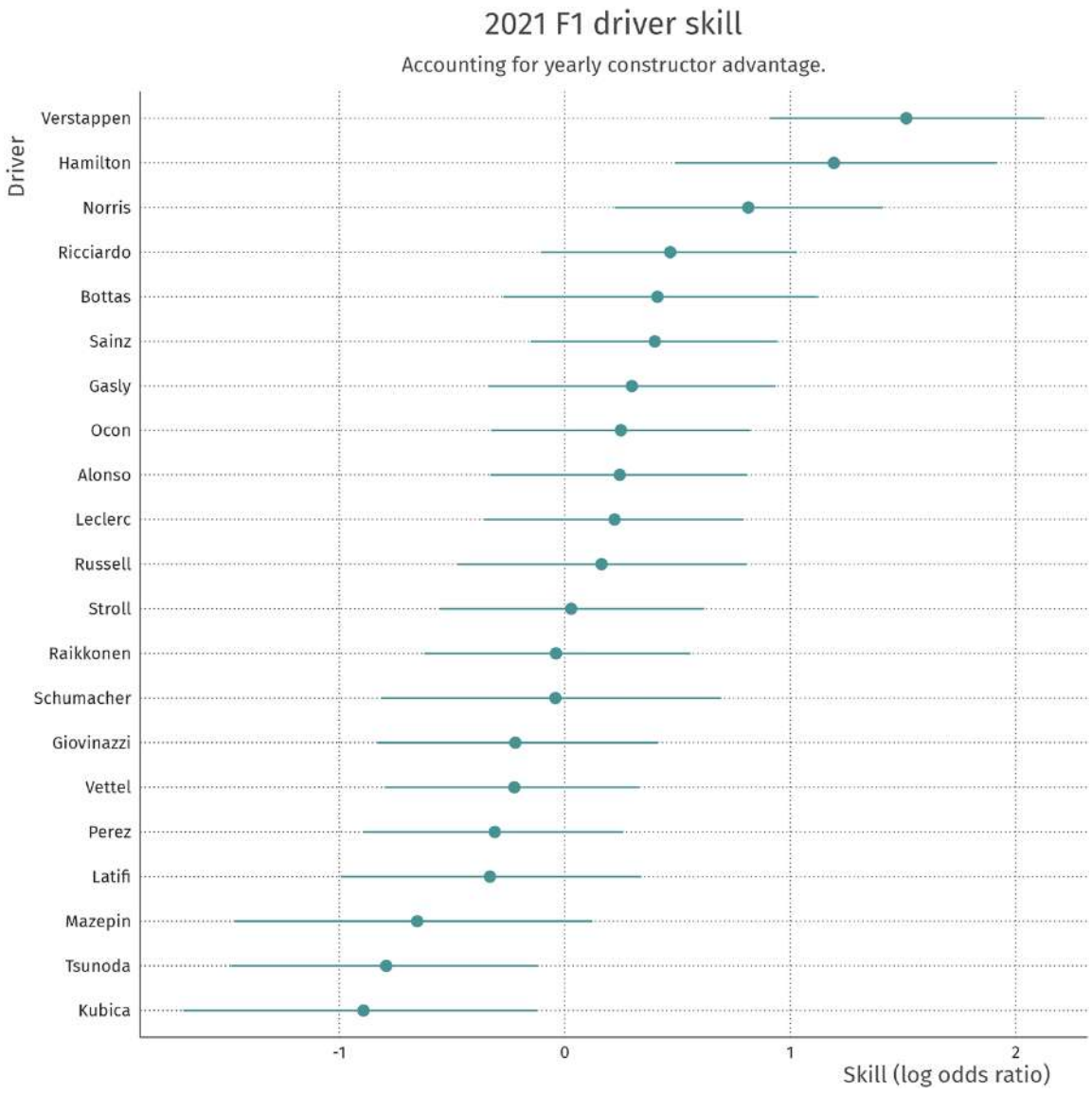


Figure 4: Driver skill in 2021 on the log odds ratio scale, accounting for yearly driver skill and constructor form. Error bars indicate 89 % credible interval.

4.3 Season performance

By combining driver skill and constructor advantage for each entrant in the 2021 season, we can create a posterior predictive simulation for the entire season. Here, we simulate finish positions (for the races where the entrant actually finished), which we then translate to points using a slightly simplified version of the 2021 points system (i.e., excluding fastest lap points and sprint race points). From this, we compute the expected average points per race over the whole 2021 season, as well as its 89 % credible interval. Then, we compare this to the realized average points for each entrant using the same simplified points system. The result of this is shown in Table 2.

Generally, the expected and observed columns line up well in terms of credible interval coverage, although the model tends to underestimate points at the top and overestimate points at the bottom. This may be due to the regularizing effects of the multilevel model implementation, both on average performance and seasonal form.

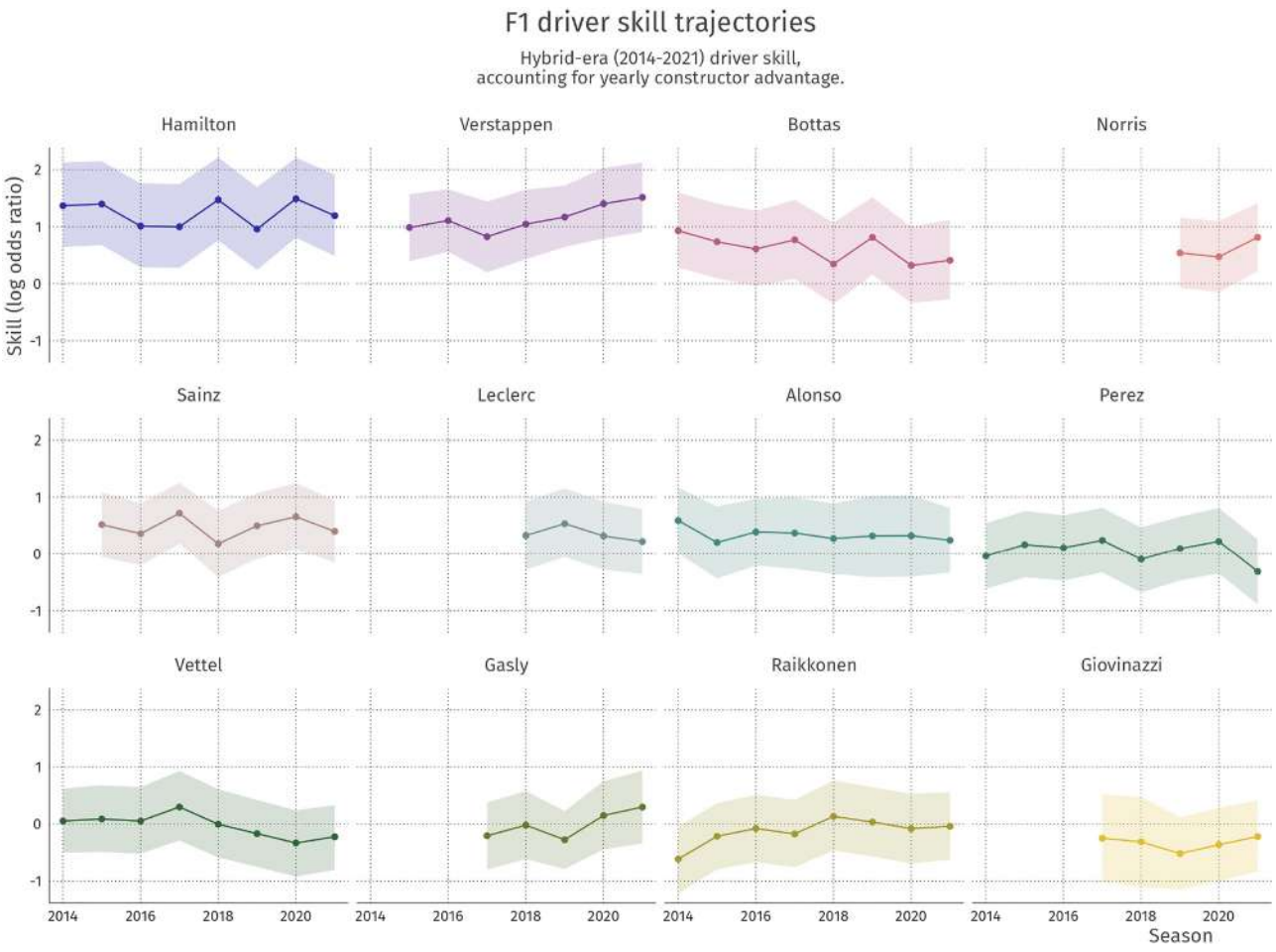


Figure 5: Driver skill trajectories for 12 drivers in the hybrid era on the log odds ratio scale. Ribbons indicate 89 % credible interval.

4.4 Relative contributions of drivers and constructors

In order to investigate the contributions of drivers and constructors to the race results, we investigate the standard deviations of the random intercepts in the model. By investigating these standard deviations, we can make conclusions about which matters more in terms of race results: the driver or the car. The posteriors for the variation coefficients are shown in Table 3.

The standard deviation of the constructor is larger than that for the driver. This means that on average, the constructor has a larger impact on race results than the driver. Rephrasing this, on average the correlation in the outcome is stronger for two different drivers driving for the same team than for the same driver driving for different teams. This interpretation is also exemplified by the race results shown in Figure 1: Räikkönen driving for Alfa Romeo (2020) looks more like his teammate (Giovinazzi) than like Räikkönen driving for Ferrari (2018).

Quantifying the relative importance of long-term constructor advantage compared to driver skill is also possible directly from the numerical summaries. The posterior estimates for the variances are as follows: $\sigma_c^2 \approx 2.65$, $\sigma_{cs}^2 \approx 0.54$, $\sigma_d^2 \approx 0.29$, and $\sigma_{ds}^2 \approx 0.12$. Following the methodology of Bell et al. (2016, §4.2), this means that constructor effects account for around 88 % of the variance in the model (89 % CI [0.775, 0.945]), which is very similar to the aforementioned authors who reported 86 %.

4.5 Counterfactual inference

Using samples from the posterior distributions of the parameters, we can answer some counterfactual questions about the drivers in the model. An example question would be: “According to the model, would Hamilton be expected to

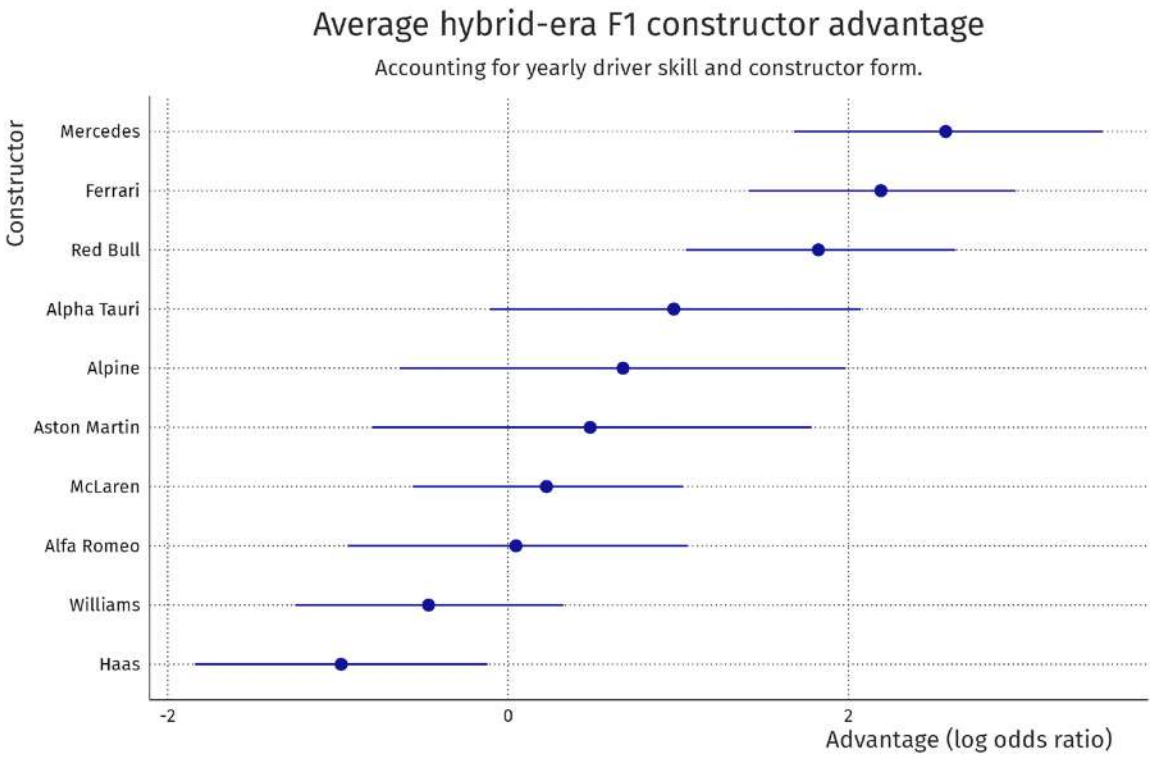


Figure 6: Long-run average constructor advantage on the log odds scale. Error bars indicate 89 % credible interval.

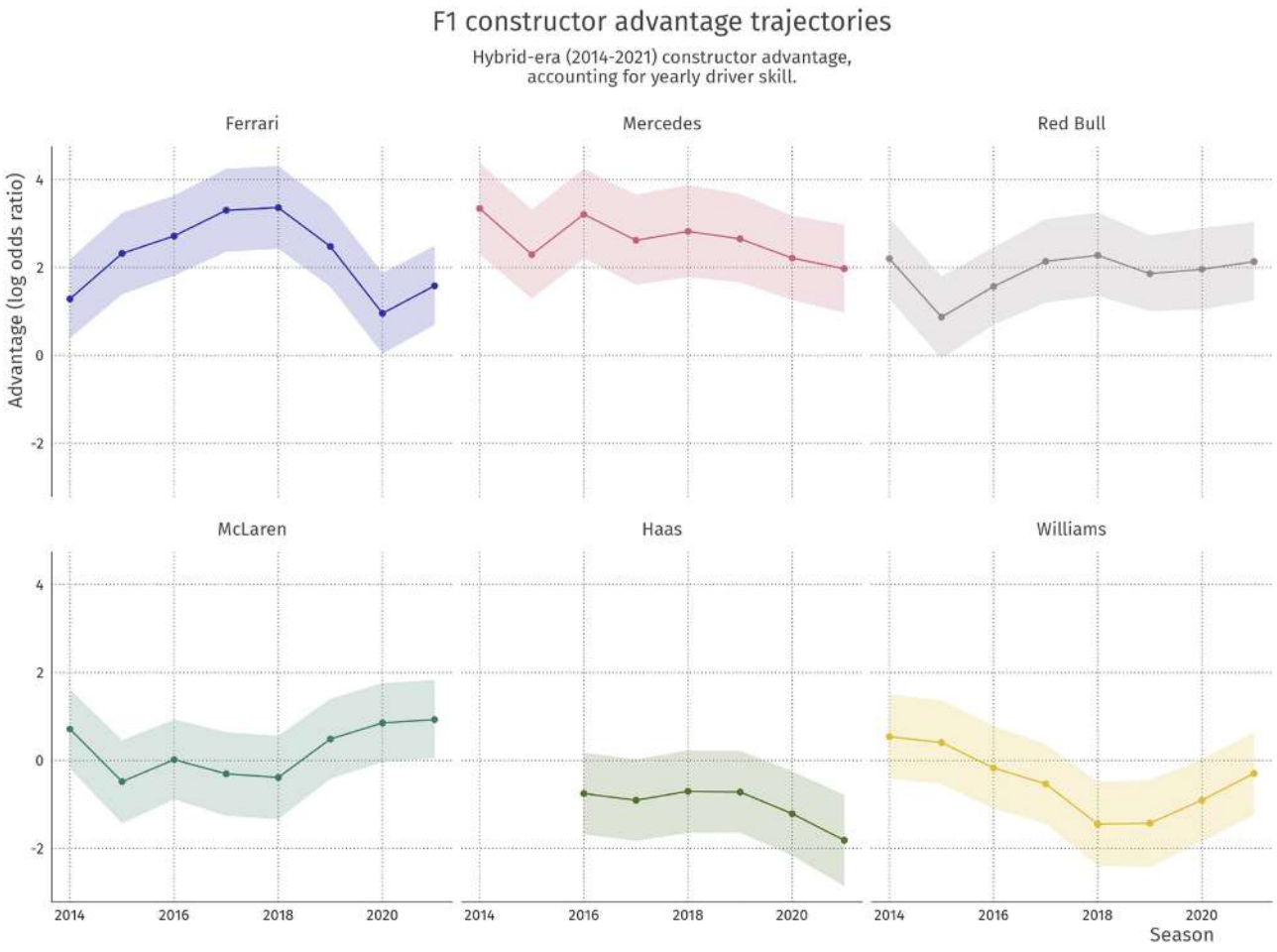


Figure 7: Yearly constructor advantage (summing constructor and constructor-year random effects) for Ferrari, Mercedes, Red Bull, McLaren, Haas, and Williams in the period 2014–2020. Ribbons indicate 89 % credible interval.

Table 2: For entrants in the 2021 season, the model's posterior expectations and 89 % credible interval for expected points per race (averaged over the 2021 season), and the realized 2021 average points per race (excluding points for fastest laps). Cases where the observed average points are outside of the 89 % CI are underlined in the last column.

Driver	Constructor	Expected points [89 % CI]	Observed points
Verstappen	Red Bull	15.30 [12.32, 18.00]	18.00
Hamilton	Mercedes	14.58 [11.45, 17.50]	17.59
Bottas	Mercedes	8.85 [5.95, 11.73]	9.82
Sainz	Ferrari	8.79 [5.73, 11.91]	7.45
Norris	McLaren	7.07 [4.23, 10.18]	7.23
Leclerc	Ferrari	7.00 [4.18, 9.95]	7.32
Perez	Red Bull	6.75 [4.00, 9.73]	8.59
Ricciardo	McLaren	5.71 [3.09, 8.64]	5.41
Gasly	Alpha Tauri	5.40 [2.86, 8.14]	5.14
Alonso	Alpine	4.13 [1.95, 6.73]	3.68
Ocon	Alpine	3.95 [1.82, 6.50]	3.50
Stroll	Aston Martin	2.61 [0.82, 4.86]	1.55
Tsunoda	Alpha Tauri	2.30 [0.68, 4.32]	1.45
Vettel	Aston Martin	2.08 [0.55, 4.09]	2.18
Raikkonen	Alfa Romeo	1.93 [0.45, 3.82]	0.45
Russell	Williams	1.26 [0.14, 2.86]	1.14
Giovinazzi	Alfa Romeo	1.95 [0.45, 3.86]	0.14
Latifi	Williams	0.80 [0.00, 2.14]	0.36
Schumacher	Haas	0.28 [0.00, 1.14]	0.00
Mazepin	Haas	0.12 [0.00, 0.73]	0.00

Table 3: Standard deviations (σ) for the random effects in the model. Lower and upper represent bounds of the 89 % credible intervals.

Component	Symbol	Estimate	Est. error	Lower	Upper
Constructor advantage	σ_c	1.63	0.34	1.14	2.27
Constructor form	σ_{cs}	0.73	0.10	0.57	0.91
Driver skill	σ_d	0.54	0.12	0.35	0.76
Driver form	σ_{ds}	0.35	0.06	0.24	0.46

beat Räikkönen in a race in 2021 if Hamilton drove for Alfa Romeo and Räikkonen for Mercedes?".

We can answer this question by computing the posterior probability of Hamilton beating Räikkönen $\pi_{\text{ham>rai}}$:

$$\begin{aligned}\theta_{\text{ham:alfa:2021}} &= \theta_{\text{ham}} + \theta_{\text{ham:2021}} + \theta_{\text{alfa}} + \theta_{\text{alfa:2021}} \\ \theta_{\text{rai:merc:2021}} &= \theta_{\text{rai}} + \theta_{\text{rai:2021}} + \theta_{\text{merc}} + \theta_{\text{merc:2021}} \\ \pi_{\text{ham>rai}} &= \frac{\exp(\theta_{\text{ham:alfa:2021}})}{\exp(\theta_{\text{ham:alfa:2021}}) + \exp(\theta_{\text{rai:merc:2021}})} \end{aligned} \tag{6}$$

The posterior of $\pi_{\text{ham>rai}}$ is shown in Figure 8. It is shown that Räikkönen is expected to beat Hamilton in this scenario ($E[\pi_{\text{ham>rai}}] < 0.5$), with a clear degree of uncertainty. Note that this counterfactual prediction is a way to summarise the model, meaning the same assumptions that accompany the model also accompany the counterfactual predictions. For example, for these predictions the data

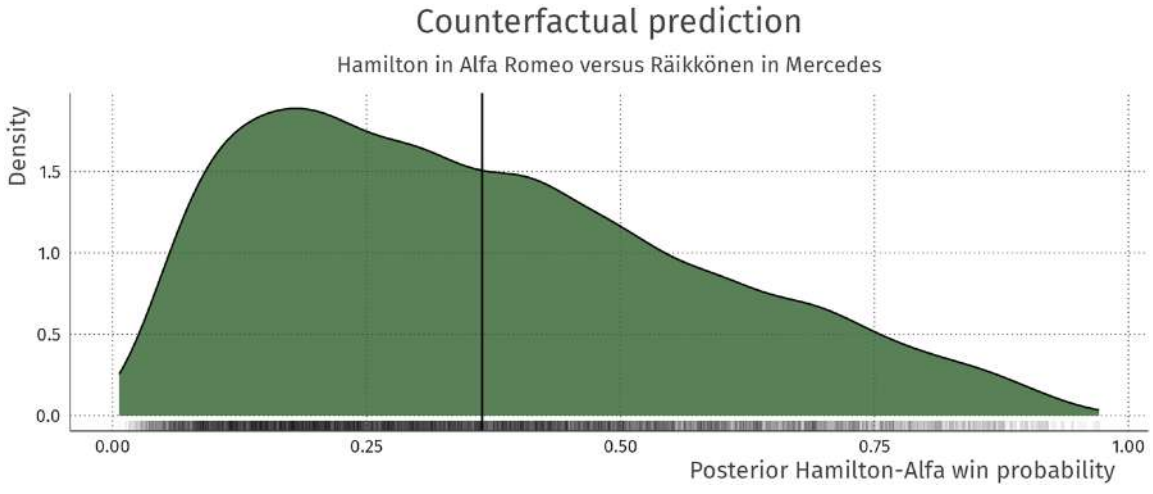


Figure 8: Posterior distribution of the win probability of Hamilton in an Alfa Romeo over Räikkönen in a Mercedes in a race during the 2021 season. The expected value of the distribution (vertical line) is 0.36, meaning that Räikkönen is expected to beat Hamilton in this scenario.

from before 2014 is irrelevant, and driver talent is independent of the constructor advantage.

5 Discussion

In this paper, we used a Bayesian multilevel rank-ordered logit model for the rank results in Formula 1 races of the hybrid era (2014–2021). After model development, comparison with leave-one-out cross-validation, and validation with posterior predictive checks, we have made inferences about the drivers and constructors competing in the 2021 season. In terms of the drivers, Hamilton and Verstappen are the best drivers. In terms of the constructors, the top three teams (Mercedes, Ferrari, and Red Bull) clearly outperform the rest on average in the hybrid era. Additionally, the model accurately represents changes in constructors' seasonal form, for example reproducing a drop in performance by Ferrari in 2020. Comparing driver contributions to team contributions, we have concluded that the car is more important than the driver when it comes to race results. Using our model and posterior sampling, it is possible to obtain answers of counterfactual questions as posterior distributions, which we have shown in the last part of Section 4.

In terms of parameter interpretation, there is an interesting parallel between this model and Elo ratings (Elo 1978) in chess. Both the Elo rating and our driver talent parameters can be transformed using an inverse logit to compare the relative strength of the competitors, and thus how likely it is that one competitor wins. The larger the difference between the ratings, the more certain it is that the competitor with the higher rating wins. We have shown such a comparison in a counterfactual situation in Section 4. Note that this Bayesian hierarchical approach has been applied before to different sports (e.g., in tennis; Ingram 2019). However, in our model not only the athletes get ratings, but also the constructors, so comparisons can be made at this level as well. This approach could be used in other sports where multiple independent components contribute to competition results.

While this model does well at describing past data, for example closely reproducing the ranking of the 2021 season, it is probably not suitable for prediction. It uses very limited information (only the driver, constructor, and year) and for each year a specific effect needs to be estimated. Before a season starts, there is no data on that season, meaning that these year-effects are unavailable (even though they are important components of this model). For forecasting, approaches such as that of Henderson et al. (2018) may be more suitable relative to the baseline of bookmaker odds.

There are several areas where this model may be improved. One area is in team continuity: teams can officially change their name, when behind the scenes it is the same team, with the same long-run performance. For example, Alpha Tauri is a re-branding of the Italian Toro Rosso team, but it enters our dataset as a completely new team, with understandably large credible intervals around performance in 2020. By not accounting for team continuity across different team names, we had in total 17 different constructors in the dataset. On the other hand, team name changes often do go hand-in-hand with some structural changes, and it is hard to determine the extent to which this happens: where to draw the line between a "re-brand" and a new team?

While our model accurately represents driver and constructor performances in the seasons 2014–2020, the data range could be expanded in order to provide results that better reflect the careers of certain drivers. For example, Räikkönen – who is at the end of his career – is in the bottom half in Figure 4 and Table 2. Because the biggest part of Räikkönen's career is missing from the data, his 2007 world championship has not been appropriately taken into account. These errors propagate, perhaps overestimating the performance of Alfa Romeo (Räikkönen's team for that year) and underestimating the skill of Giovinazzi (whose only teammate has been Räikkonen).

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: None declared.
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

Appendix A: Sensitivity analysis for race finishing

In the main paper, we remove non-finishers from the data, estimating the model parameters data only from the races in which a certain competitor finished the race. This choice influences the interpretation of the parameters: the skill parameters exclude any measure of "reliability" because only finished races count. In this section, we perform sensitivity analysis for this choice, investigating the effects on the results of making a different choice here:

1. Not removing any results, i.e., ranking unfinished competitors based on race distance.
2. Removing only car-related non-finishes and ranking the driver-related non-finishes as normal.

3. Removing only driver-related non-finishes and ranking the car-related non-finishes as normal.
4. Only ranking finished competitors (i.e., original analysis from main paper).

As mentioned in the main text in Section 2, yet another option is to include non-finishes in the model directly, resulting in a joint binomial model for finishing and rank model for ranking conditional on finishing. This approach is outside the scope of this sensitivity analysis.

In the data, it is difficult to accurately ascertain which results belong to car-related and driver-related errors: there are 51 different status types other than finishing the race; some are ambiguous with respect to who caused them, e.g., "Excluded", "Technical", "Damage". For the purpose of this sensitivity analysis, we have made the following selection:

Driver-related: Collision, Disqualified, Withdrew, Retired, Accident, Collision damage, Spun off, Excluded, Illness.

Car-related: ERS, Oil pressure, Engine, Technical, Gearbox, Electrical, Power Unit, Brakes, Clutch, Exhaust, Mechanical, Turbo, Rear wing, Drivetrain, Suspension, Oil leak, Water leak, Water pressure, Electronics, Transmission, Wheel, Power loss, Fuel system, Front wing, Tyre, Throttle, Brake duct, Hydraulics, Battery, Puncture, Overheating, Wheel nut, Vibrations, Driveshaft, Fuel pressure, Seat, Spark plugs, Steering, Damage, Out of fuel, Debris, Radiator.

The results of the sensitivity analysis are shown graphically in Figures 9 and 10 (driver skills) and 11 (constructor advantage). A general pattern in comparing the main paper model ("only finishers retained") to the other three models is the smaller variance of these random effect components; when we include non-finishes in the ranking data, this ranking becomes more noisy so the latent skill and advantage parameters will be closer together (and closer to 0). This is especially visible in Figure 11, which was faceted by the model variable for this purpose.

For the drivers, generally the skill patterns over time (Figure 9) are similar across different models, meaning that skill increases and decreases are captured somewhat similarly for the different models. Looking at the skill level in 2021, we see that a notable outlier is Bottas, who drops dramatically in skill level relative to his surrounding drivers once non-finishes are included in the model. It is unclear

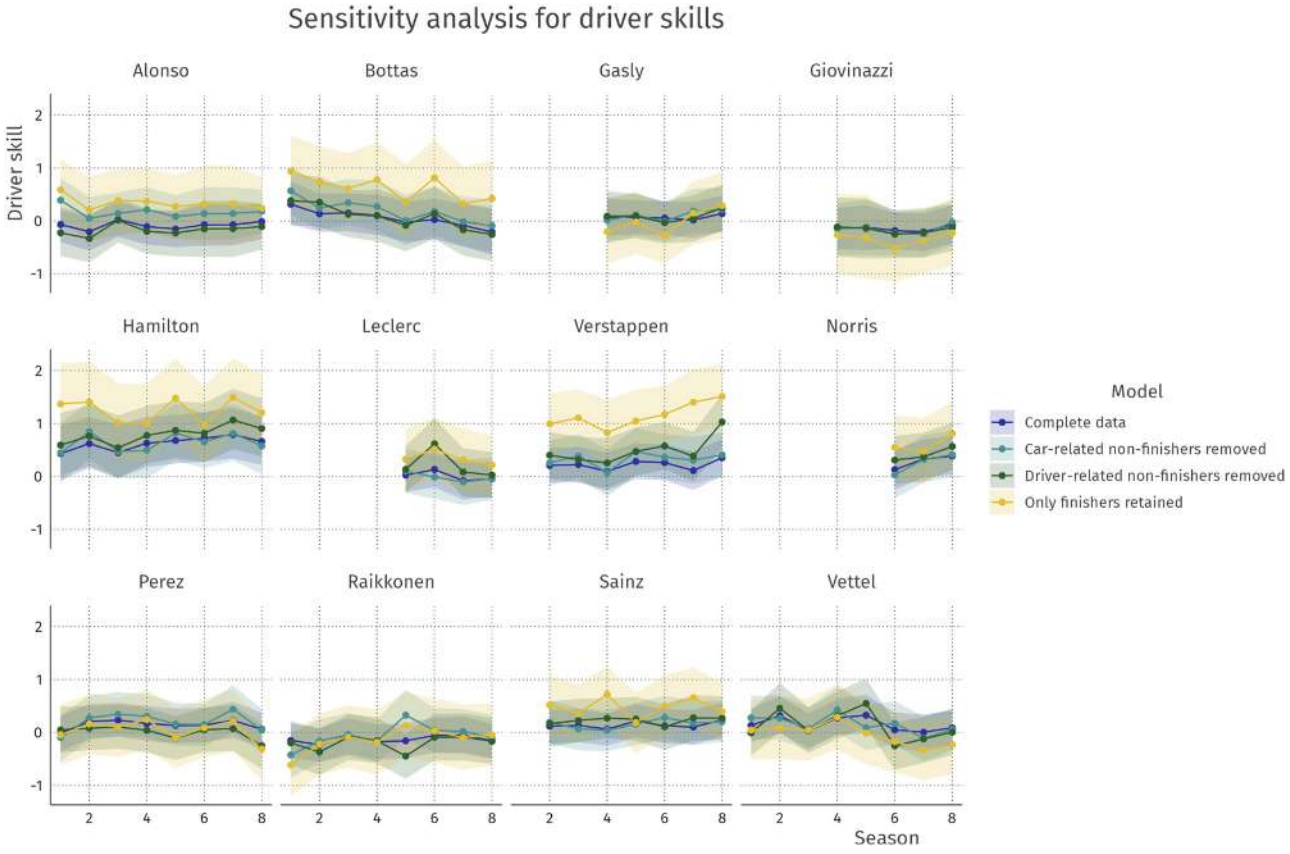


Figure 9: Sensitivity analysis for driver skills inferences.

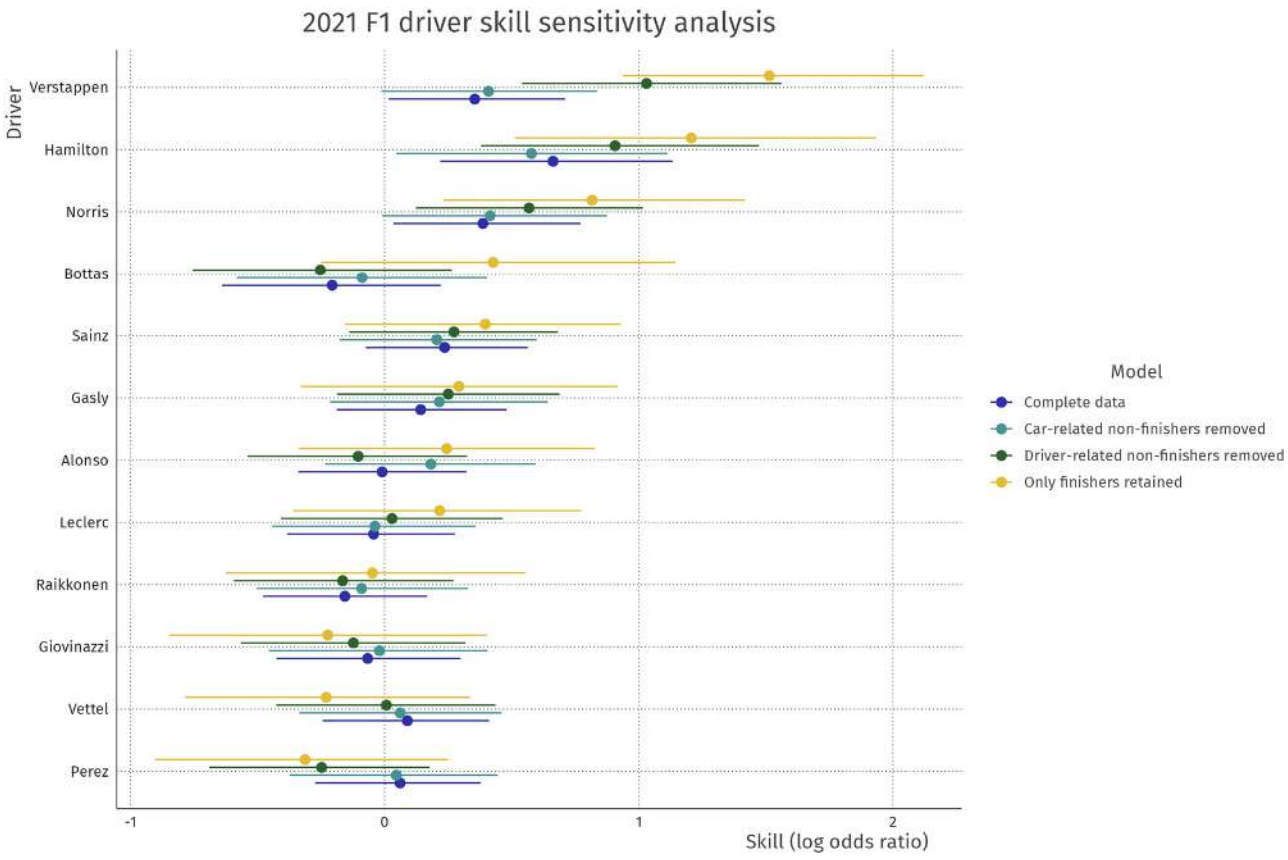


Figure 10: Sensitivity analysis for driver skills inferences for the 2021 season.

what the mechanism is behind this drop, as Bottas has not had much more non-finishes than other drivers (4 per season, whereas the average is 3).

In summary, the results are somewhat sensitive to the choice of what to include in the model. The conclusions in the main paper should be interpreted carefully; we once again make clear that the interpretation of the skill parameters only includes *finished* races, thus do not contain any component of reliability of the car or carefulness of the driver. For example, Pastor Maldonado — who was notoriously crash-prone — is ranked 6th worst if we include all data in the model, but a much better 19th of 38 drivers if we only look at finished races.

Appendix B: Extended model definition

In this section, we extend the model definition in Section 3 in the main paper into a fully specified generative model. Following Glickman and Hennessy (2015), the Rank-Ordered Logit (ROL) distribution implies that the skills for each competitor in the rth race ($\boldsymbol{\vartheta}_r$) generate a latent *performance* vector $\boldsymbol{z}_r$ following independent standard extreme value (Gumbel) distributions. The resulting rank-ordering $\boldsymbol{y}_r$ is the rank-ordering of these latent performance values. By replacing RankOrderedLogit by this generative process, the full model can be specified as follows:

$$
\begin{aligned}
\boldsymbol{y}_r &= \text{rank}(\boldsymbol{z}_r) \\
\boldsymbol{z}_r &= \{z_c \mid c \in C_r\} \\
z_c &\sim \text{Gumbel}(\vartheta_c) \\
\vartheta_c = \theta_{dts} &= \theta_d + \theta_{ds} + \theta_t + \theta_{ts} \\
\theta_d &\sim \mathcal{N}\left(0, \sigma_d^2\right) \\
\theta_{ds} &\sim \mathcal{N}\left(0, \sigma_{ds}^2\right) \\
\theta_t &\sim \mathcal{N}\left(0, \sigma_t^2\right) \\
\theta_{ts} &\sim \mathcal{N}\left(0, \sigma_{ts}^2\right)
\end{aligned} \tag{7}
$$

where we make use of the standard Gumbel distribution with only a location parameter μ and no scale:

$$
f(x; \mu) = \exp\left(x - \mu - \exp(x - \mu)\right) \tag{8}
$$

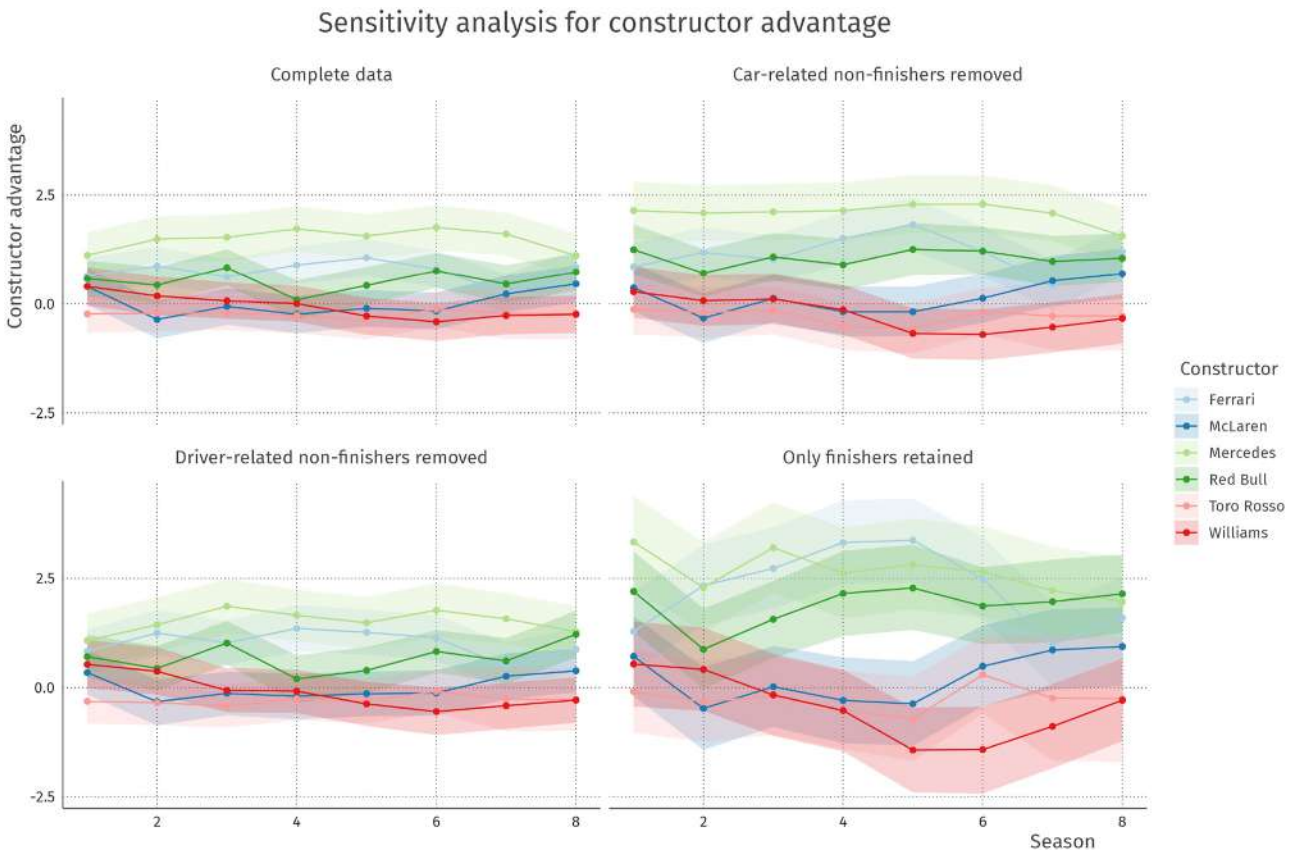


Figure 11: Sensitivity analysis for constructor advantage inferences.

The likelihood for the model in Equation (7) can be simplified for implementation (Glickman and Hennessy 2015, Equation (4)). Assuming, without loss of generality, that the set of competitors for a specific race C_r is already ordered according to the race result $\boldsymbol{y}_r$, we obtain the following likelihood:

$$p(\boldsymbol{y}_r|\boldsymbol{\vartheta}_r) = p(z_1 > z_2 > \ldots > z_{m_r}|\boldsymbol{\vartheta}_r) = \prod_{i=1}^{m_r-1} \frac{\exp(\vartheta_i)}{\sum_{j=i}^{m_r} \exp(\vartheta_j)} \tag{9}$$

where m_r is the number of competitors $|C_r|$.

Appendix C: Traceplot

Figure 12 shows the traceplot for the main standard deviation parameters of the rank-ordered logit model. From top to bottom, the standard deviations for the four random effects β_d, β_{ds}, β_c, and β_{cs} are shown. All traceplots for the four different chains overlap as expected from an appropriately converged model.

Appendix D: Comparing different dynamic skill parameter implementations

In the paper, the model does not take time into account explicitly, but it estimates i.i.d. seasonal form parameters per driver and constructor. Here, this approach is briefly compared to explicitly using a latent multilevel AR(1) implementation (`auto`), and to a more parsimonious multilevel intercept + slope model (`slope`) for the latent skill parameters. For the auto-regressive model, we used a cluster mean centered parameterization as in Hamaker and Grasman (2015).

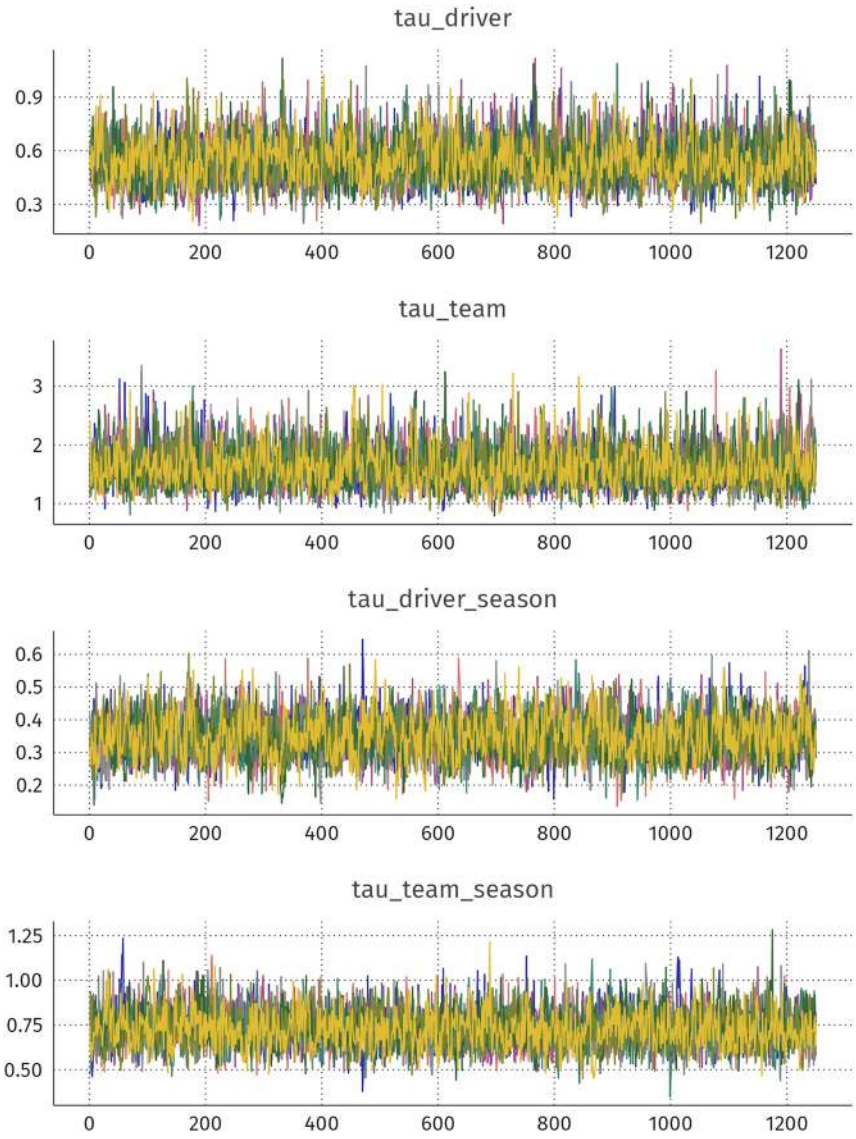


Figure 12: Monte Carlo markov chain visualisation for the main model parameters, indicating satisfactory convergence for all shown parameters.

Because these models have the same likelihood form (the rank-ordered logit likelihood), we can compare them using efficient leave-one-out cross-validation:

	ELPD	SE_{ELPD}	Δ	SE_{Δ}
Auto	−3992.7	83.9		
Rank	−3993.8	83.9	−1.1	1.7
Slope	−4042.9	81.6	−50.2	18.4

The simple slope model is quite clearly worse than the other two implementations (Figure 13); it is not able to capture the skills properly. For example, looking at Giovinazzi and Räikkönen in the plot below shows that they are estimated to be top drivers in the 2021 season, which is unlikely given their results.

The AR(1) and default rank model are very similar in terms of LOOCV performance. Looking at the estimated skill trajectories, it becomes clear why this is the case (Figure 14).

The driver skills have very similar estimates. The same holds for the constructor advantage over time (Figure 15).

Based on this comparison, the decision was made to work with the more interpretable basic multilevel model in the main paper.

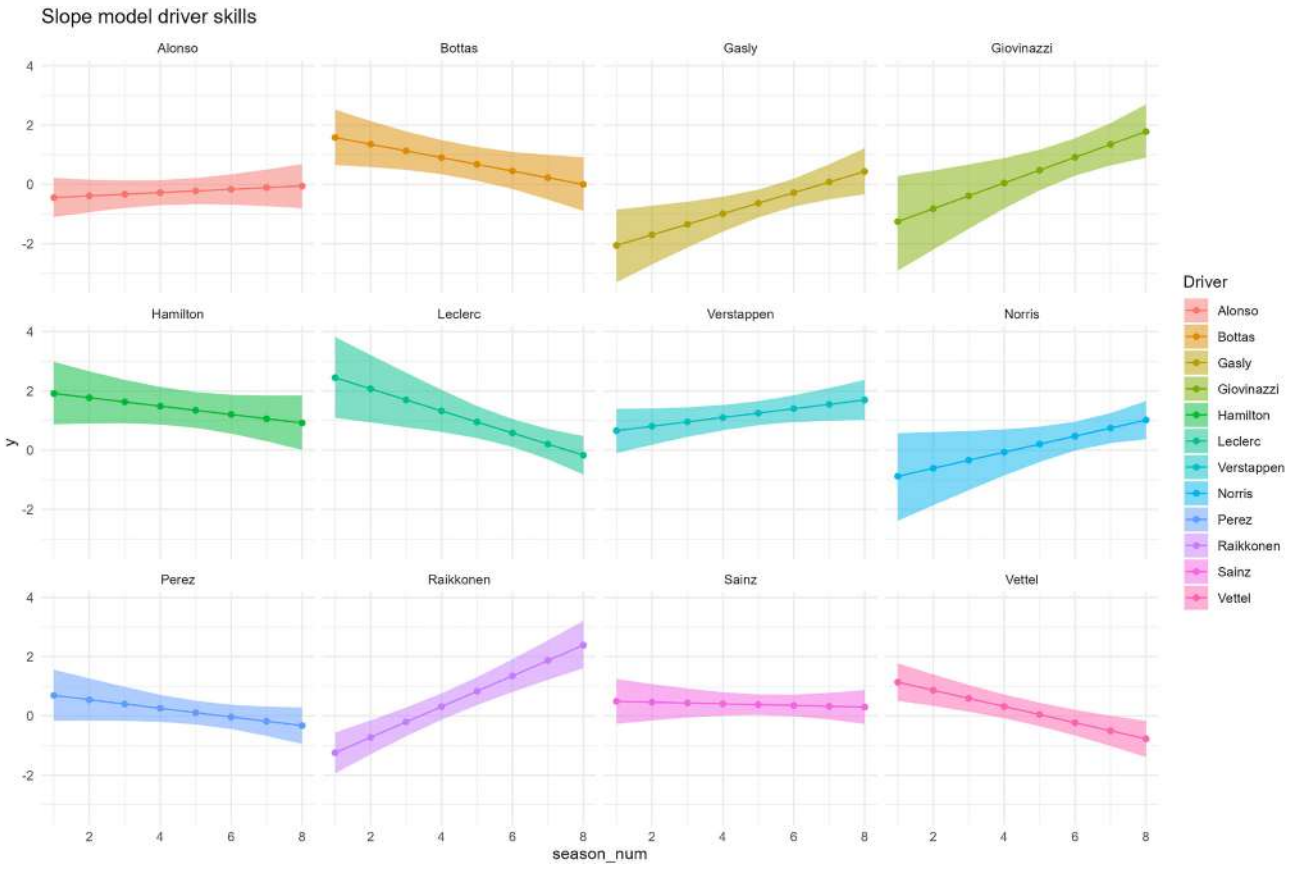


Figure 13: Posterior distribution of driver skill parameters over seasons for the slope model. Ribbons indicate 89 % credible interval.

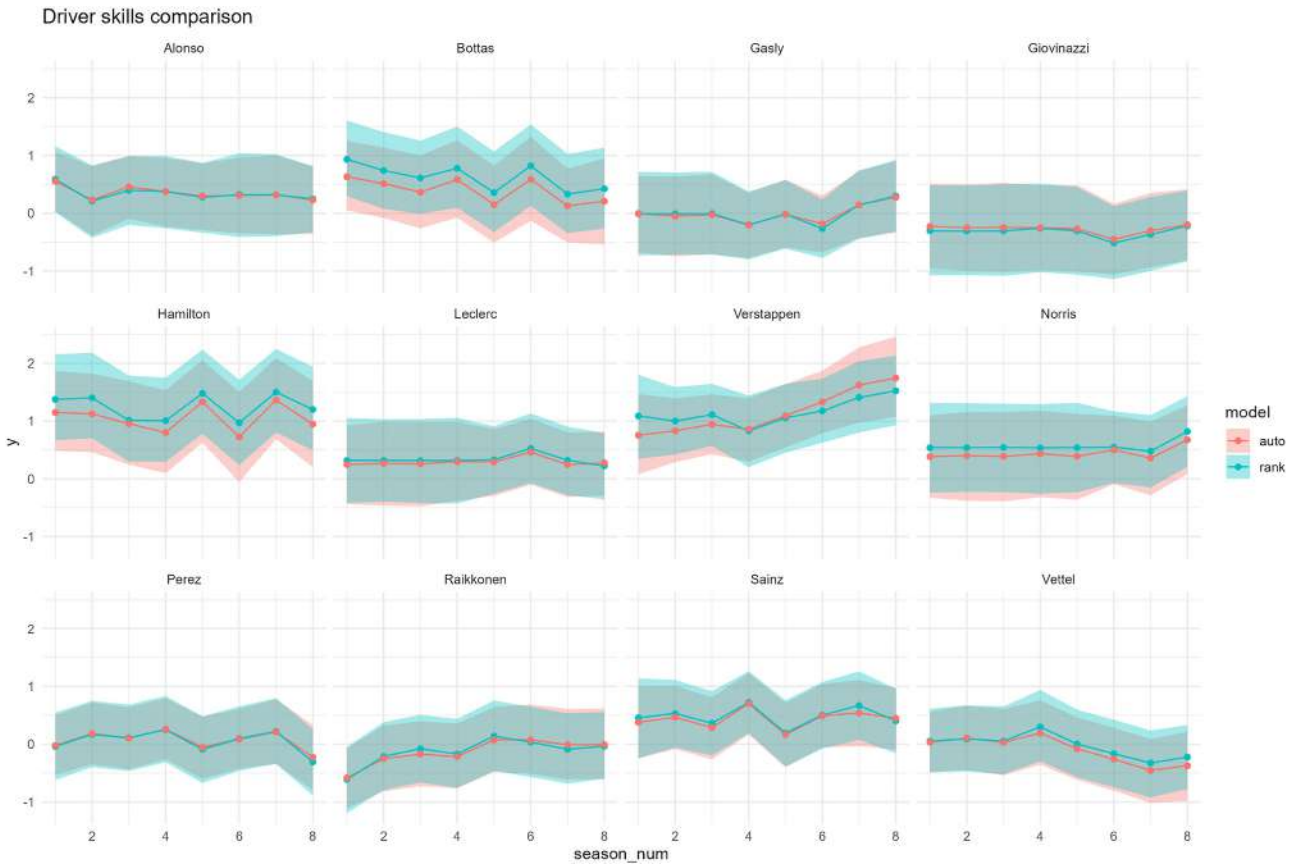


Figure 14: Posterior distribution of driver skill parameters over seasons for the multilevel and AR(1) implementations. Ribbons indicate 89 % credible interval.

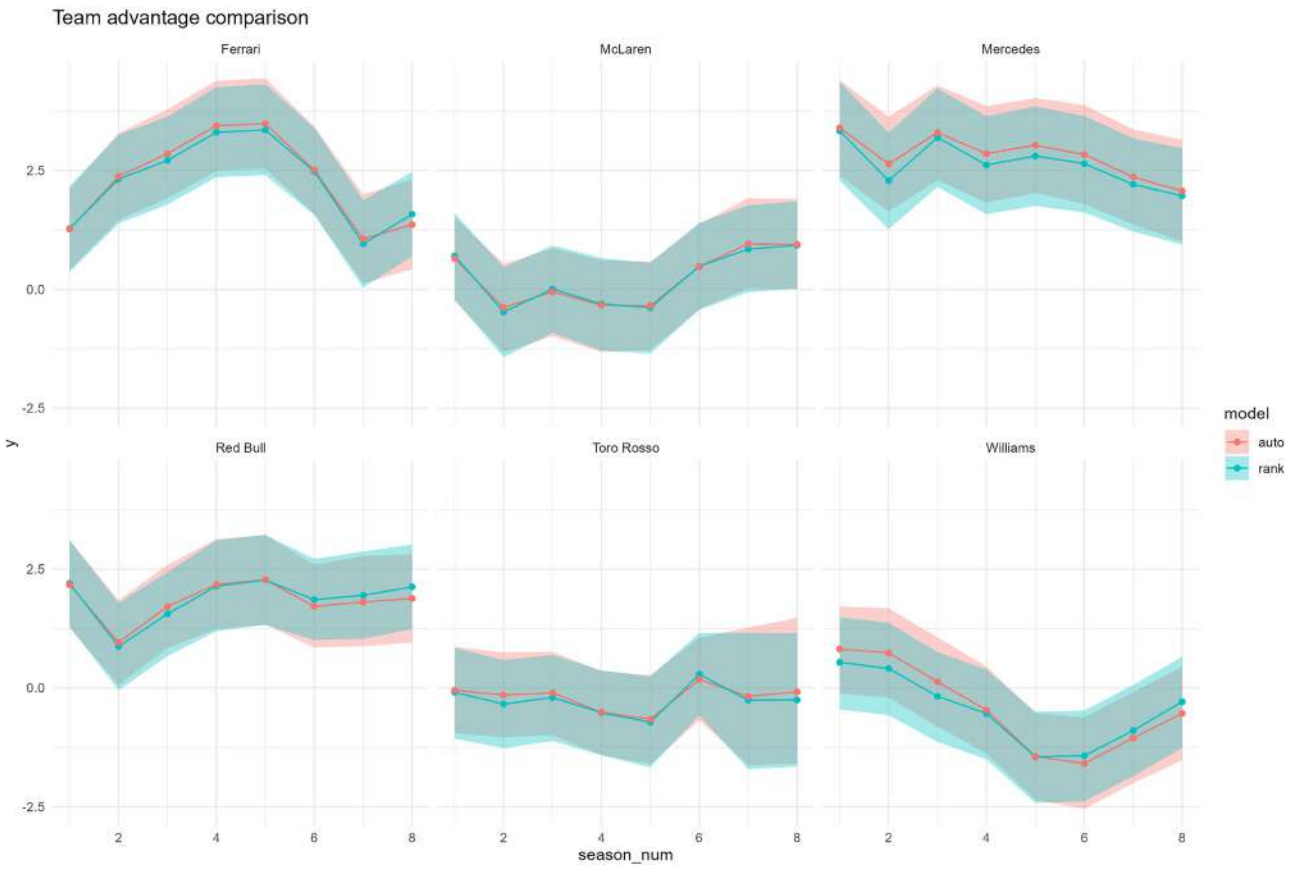


Figure 15: Posterior distribution of team advantage parameters over seasons for the multilevel and AR(1) implementations. Ribbons indicate 89 % credible interval.

References

Bell, A., J. Smith, C. E. Sabel, and K. Jones. 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.

Bol, R. 2020. "How to Win in Formula One: Is it the Driver or the Car?" In *The Correspondent*. Also available at: https://thecorrespondent.com/642/how-to-win-in-formula-one-is-it-the-driver-or-the-car.

Budzinski, O., and A. Feddersen. 2020. "Measuring Competitive Balance in Formula One Racing." In *Outcome Uncertainty in Sporting Events*. Cheltenham: Edward Elgar Publishing.

Bürkner, P.-C. 2017. "Brms: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software* 80 (1): 1—28.

Carpenter, B., A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. 2017. "Stan: A Probabilistic Programming Language." *Journal of Statistical Software* 76 (1): 1—32.

Eichenberger, R., and D. Stadelmann. 2009. "Who Is the Best Formula 1 Driver? an Economic Approach to Evaluating Talent." *Economic Analysis and Policy* 39 (3): 389—406.

Elo, A. 1978. *The Rating of Chess Players, Past and Present*. New York: Arco.

Formula1.com, 2020. *Fia Reaches 'settlement' with Ferrari Following 2019 Engine Investigation*. Also available at: https://www.formula1.com/en/latest/article.fia-reaches-settlement-with-ferrari-following-2019-engine-investigation.6beur1atKeTLvJHPEuHUJW.html (accessed 1 May, 2021).

Gabry, J., D. Simpson, A. Vehtari, M. Betancourt, and A. Gelman. 2019. "Visualization in Bayesian Workflow." *Journal of the Royal Statistical Society: Series A* 182 (2): 389—402.

Gelman, A. 2006. "Multilevel (Hierarchical) Modeling: What it Can and Cannot Do." *Technometrics* 48 (3): 432—5.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. 2013. *Bayesian Data Analysis*. Boca Raton: CRC Press.

Glickman, M. E., and J. Hennessy. 2015. "A Stochastic Rank Ordered Logit Model for Rating Multi-Competitor Games and Sports." *Journal of Quantitative Analysis in Sports* 11 (3): 131—44.

Hamaker, E. L., and R. P. Grasman. 2015. "To Center or Not to Center? Investigating Inertia with a Multilevel Autoregressive Model." *Frontiers in Psychology* 5: 1492.

Henderson, D. A., and L. J. Kirrane. 2018. "A Comparison of Truncated and Time-Weighted Plackett—Luce Models for Probabilistic Forecasting of Formula One Results." *Bayesian Analysis* 13 (2): 335—58.

Ingram, M. 2019. "A Point-Based Bayesian Hierarchical Model to Predict the Outcome of Tennis Matches." *Journal of Quantitative Analysis in Sports* 15 (4): 313—25.

Ingram, M. 2021. *A First Model to Rate Formula 1 Drivers*. Also available at: https://martiningram.github.io/f1-model/ (accessed 1 March, 2022).

McElreath, R. 2018. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Boca Raton: Chapman and Hall/CRC.

Newell, C. 2021. *Ergast Developer API*. Also available at: http://ergast.com/ (accessed 1 February, 2022).

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—78.

Van Der Maas, H. L., and E.-J. Wagenmakers. 2005. "A Psychometric Analysis of Chess Expertise." *American Journal of Psychology* 118 (1): 29—60.

van Kesteren, E.-J., and T. Bergkamp. 2023. "Vankesteren/f1Model: Rank-Ordered Logit Model." *Zenodo*. https://doi.org/10.5281/zenodo.7632045.

Vehtari, A., A. Gelman, and J. Gabry. 2017. "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and Waic." *Statistics and Computing* 27 (5): 1413—32.

Supplementary Material: This article contains supplementary material (https://doi.org/10.1515/jqas-2022-0021).

Advanced Machine Learning Approaches for Formula 1 Race Performance Prediction: A Comprehensive Analysis of Championship Point Forecasting

Aayam Bansal*, Aadit Arora†, Lakshay Bhati‡, Kushagra Sethia§,
Ishani Verma¶, Naisha Kapoor‖
*aayam@levitas.in, †aadit@levitas.in, ‡lakshay@levitas.in, §kushagra@levitas.in,
¶ishani@levitas.in, ‖naisha@levitas.in
Levitas, Amity International School, Sec - 46, Gurgaon

***Abstract*—This paper presents a comprehensive machine learning framework for predicting Formula 1 race performance and championship point allocation using an extensive dataset spanning 74 years of racing history from 1950 to 2024. Our methodology encompasses the analysis of 589,081 individual lap times across 1,125 races, incorporating multiple algorithmic approaches including ensemble methods, gradient boosting techniques, and traditional regression models. The research employs sophisticated feature engineering strategies to extract meaningful predictors from qualifying performance, lap time variations, circuit characteristics, and temporal racing dynamics. Our optimal model, utilizing Gradient Boosting algorithms, achieved exceptional predictive accuracy with an R^2 score of 0.999, RMSE of 0.197, and MAE of 0.125. Comprehensive feature importance analysis revealed that race position contributes 75.8% to prediction accuracy, followed by seasonal variations at 23.8%. Cross-validation experiments demonstrate robust model generalization with a mean R^2 of 0.993 ± 0.013 across multiple data partitions. This research significantly advances sports analytics methodologies and provides practical applications for Formula 1 teams, broadcasters, and strategic decision-making processes.**

***Index Terms*—Formula 1, Machine Learning, Predictive Analytics, Sports Analytics, Gradient Boosting, Performance Prediction, Championship Forecasting, Ensemble Methods**

I. Introduction

Formula 1 represents the pinnacle of motorsport technology and data-driven competition, generating unprecedented volumes of telemetry data, performance metrics, and strategic information that provide unique opportunities for advanced analytical modeling [1], [2]. The sport's evolution from mechanical engineering excellence to data science sophistication has created an ideal environment for applying cutting-edge machine learning techniques to predict race outcomes and championship point distributions [3], [4].

The complexity inherent in F1 racing stems from the intricate interplay of numerous variables including driver expertise, vehicle aerodynamics, power unit performance, tire strategies, weather conditions, circuit characteristics, and real-time strategic decisions made during race events [5], [6]. Traditional statistical approaches have proven insufficient for capturing these multifaceted interactions, necessitating the development of sophisticated machine learning methodologies capable of modeling non-linear relationships and temporal dependencies [7], [8].

Contemporary F1 teams invest heavily in predictive analytics to gain competitive advantages, optimize resource allocation, and enhance strategic decision-making processes. The ability to accurately forecast race outcomes, predict championship point distributions, and identify key performance indicators has become crucial for team success in the modern era [9], [10]. However, existing research in this domain has been limited by dataset scope, methodological approaches, and the complexity of feature engineering required for motorsport analytics [11], [12].

This research addresses these limitations by developing a comprehensive machine learning framework that analyzes 74 years of Formula 1 historical data to create highly accurate predictive models for championship point allocation. Our approach incorporates advanced feature engineering techniques, multiple algorithmic comparisons, and rigorous validation methodologies to establish new benchmarks in motorsport analytics.

A. Research Contributions and Significance

The primary contributions of this research encompass several key areas of advancement in sports analytics and machine learning applications. First, we present the most comprehensive analysis of Formula 1 historical data ever undertaken, incorporating 589,081 individual lap times across 1,125 races from 1950 to 2024, providing unprecedented temporal coverage and statistical power for predictive modeling [**?**].

Second, our methodology introduces novel feature engineering techniques specifically designed for motorsport analytics, including temporal lap time variations, grid position deltas, and circuit-specific performance indicators that capture the unique dynamics of F1 racing [13]. These innovations enable more accurate representation of the complex factors influencing race outcomes.

Third, we provide the first systematic comparison of multiple machine learning algorithms applied to F1 prediction tasks, including traditional regression methods, ensemble approaches, and gradient boosting techniques, establishing performance benchmarks for future research [14]. Our evaluation framework incorporates cross-validation strategies and statistical significance testing to ensure robust model assessment.

Fourth, the research identifies and quantifies the relative importance of various performance indicators in F1 championship point prediction, providing valuable insights for team strategists, broadcasters, and academic researchers interested in motorsport analytics [15]. These findings contribute to the theoretical understanding of factors driving competitive success in Formula 1.

II. Literature Review and Related Work

The application of data analytics and machine learning techniques to motorsport has evolved significantly over the past two decades, with Formula 1 serving as a primary testbed for advanced analytical methodologies due to its data-rich environment and competitive intensity [16], [17]. Early research in this domain focused primarily on traditional statistical approaches and descriptive analytics, gradually evolving toward predictive modeling and machine learning applications [18].

Henderson et al. [11] conducted pioneering work in F1 performance analysis, establishing foundational relationships between qualifying positions and race outcomes using correlation analysis and basic regression modeling. Their findings demonstrated the significant impact of grid position on final race results, with correlation coefficients exceeding 0.7 in most racing scenarios. However, their approach was limited by linear assumptions and did not account for the complex interactions between multiple performance variables.

The integration of machine learning techniques into motorsport analytics gained momentum with the work of Kumar and Singh [7], who explored ensemble methods for predicting race results using decision trees and random forest algorithms. Their research demonstrated the potential of non-linear modeling approaches for capturing the complex dynamics of racing performance, achieving prediction accuracies of approximately 85% for podium finishes. Nevertheless, their study was constrained by a relatively small dataset covering only five racing seasons and limited feature engineering capabilities.

Recent advances in deep learning have opened new possibilities for motorsport analytics, with Rossi et al. [9] utilizing neural networks and recurrent architectures for lap time prediction and strategy optimization. Their deep learning models achieved significant improvements over traditional regression methods, particularly in capturing temporal dependencies and sequential patterns in racing data. However, the interpretability of these models remained limited, reducing their practical applicability for strategic decision-making processes.

The application of gradient boosting techniques to sports analytics has shown promising results across various domains [19], with XGBoost and LightGBM demonstrating superior performance in handling complex feature interactions and providing robust predictions [20], [21]. These methodologies have been successfully applied to other sports including basketball [22], soccer [23], and tennis [24], but their application to Formula 1 analytics has been limited.

Feature engineering represents a critical component of successful machine learning applications in motorsport, with previous research emphasizing the importance of domain-specific knowledge in creating meaningful predictors [25]. Temporal features, circuit characteristics, weather conditions, and strategic indicators have been identified as key components for effective F1 prediction models [26], [27].

Cross-validation and model validation strategies in sports analytics have received increasing attention, with researchers emphasizing the importance of temporal validation techniques that respect the time-series nature of sports data [28]. Traditional cross-validation approaches may lead to data leakage and overly optimistic performance estimates when applied to sequential sports data [29].

III. Methodology and Experimental Design

A. Dataset Composition and Characteristics

Our research utilizes a comprehensive Formula 1 dataset encompassing 74 years of racing history from 1950 to 2024, representing the most extensive temporal coverage in motorsport analytics literature. The dataset comprises 14 interconnected tables containing detailed information about races, drivers, constructors, circuits, lap times, qualifying sessions, and championship standings. The total dataset includes 1,125 individual races across 77 unique circuits in 35 countries, with 589,081 recorded lap times from 861 distinct drivers representing 212 different constructor teams.

The lap times dataset forms the core of our analysis, containing individual lap recordings with millisecond precision, enabling detailed analysis of performance variations throughout race events. Each lap time record includes driver identification, race context, lap number, position during the lap, and precise timing measurements. This granular data allows for sophisticated feature engineering approaches that capture the dynamic nature of F1 racing performance.

Circuit characteristics are represented through geographical coordinates, elevation data, and historical performance metrics, enabling the incorporation of track-specific factors that influence lap times and race outcomes. The circuits range from sea-level street courses to high-altitude permanent facilities, with elevations spanning from -7 meters to 2,227 meters above sea level, providing diverse environmental conditions for model training.

Driver and constructor data includes performance statistics, championship standings, and historical success metrics across multiple seasons. The temporal span of the dataset captures significant evolution in F1 regulations, technology, and competitive dynamics, requiring sophisticated modeling approaches to account for these temporal variations.

B. Data Preprocessing and Quality Assessment

Data preprocessing involved comprehensive quality assessment procedures to ensure the integrity and reliability of our analytical foundation. Missing value analysis revealed minimal data gaps, with only 0.07% missing values in the qualifying dataset and complete data availability across all other primary tables. Duplicate detection algorithms identified zero duplicate records, confirming the high quality of the source data.

Outlier detection focused on identifying anomalous lap times that could indicate data recording errors, technical failures, or exceptional circumstances. Lap times exceeding three standard deviations from the mean were flagged for individual assessment, with legitimate outliers (such as safety car periods or mechanical issues) retained with appropriate contextual annotations.

Data type optimization and memory management procedures were implemented to handle the large dataset efficiently, with appropriate encoding schemes applied to categorical variables and numerical precision optimized for computational efficiency. The resulting clean dataset maintained 99.93% of original records while ensuring analytical reliability.

C. Feature Engineering and Selection

Feature engineering represents a critical component of our methodology, incorporating domain expertise to create meaningful predictors that capture the complex dynamics of F1 racing. Our approach encompasses multiple categories of engineered features designed to represent different aspects of racing performance and strategic factors.

Temporal features include average lap times, fastest lap achievements, lap time standard deviations, and lap-to-lap variation metrics that capture consistency and peak performance characteristics. These features provide insights into driver and vehicle performance throughout race events, enabling the identification of strategic patterns and performance trends.

Positional features incorporate grid position effects, position changes during races, and grid-to-finish position deltas that quantify the impact of qualifying performance and overtaking capabilities. These features account for the strategic importance of track position in Formula 1 racing and its relationship to final race outcomes.

Circuit-specific features utilize geographical and historical data to create track characteristic indicators, including elevation categories, geographical regions, and historical performance patterns. These features enable the model to account for circuit-specific factors that influence lap times and race dynamics.

Seasonal and temporal features capture the evolution of competitive balance, regulation changes, and technological development across the 74-year dataset span. These features are essential for accounting for the significant changes in F1 competition over time and ensuring model relevance across different eras.

D. Machine Learning Algorithm Selection and Implementation

Our comparative analysis encompasses seven distinct machine learning algorithms, ranging from traditional linear methods to advanced ensemble techniques. This comprehensive approach enables the identification of optimal modeling strategies for F1 prediction tasks and provides insights into the relative effectiveness of different algorithmic approaches.

Linear regression methods, including standard, Ridge, and Lasso variants, serve as baseline models and provide interpretable relationships between features and championship points. These models offer computational efficiency and clear coefficient interpretations, making them valuable for understanding basic performance relationships.

Ensemble methods, including Random Forest, Gradient Boosting, XGBoost, and LightGBM, leverage multiple decision trees to capture complex non-linear relationships and feature interactions. These algorithms excel at handling the multifaceted nature of F1 performance prediction and provide robust predictions across diverse racing scenarios.

Hyperparameter optimization procedures were implemented using grid search and cross-validation techniques to ensure optimal model performance. Each algorithm was tuned using appropriate parameter spaces and validation strategies to maximize predictive accuracy while avoiding overfitting.

IV. Results and Analysis

A. Dataset Characteristics and Exploratory Analysis

The comprehensive exploratory analysis of our 74-year Formula 1 dataset reveals fascinating insights into the evolution and characteristics of world championship racing. The lap time distribution exhibits a mean of 95.39 seconds with a standard deviation of 57.08 seconds, reflecting the diverse nature of F1 circuits and the technological evolution of the sport. The substantial variation in lap times is attributable to the wide range of circuit configurations, from high-speed circuits like Monza to technical street circuits like Monaco, as well as the significant technological advances in vehicle performance over the seven-decade span.

Statistical analysis of the relationship between qualifying and race performance confirms the critical importance of grid position in Formula 1 success. The correlation between grid position and final race position demonstrates a strong positive relationship (r = 0.711, p ¡ 0.001), validating the strategic emphasis teams place on Saturday qualifying sessions. This relationship has remained remarkably consistent across different regulatory eras, suggesting that the fundamental importance of qualifying performance transcends specific technical regulations.

The championship points distribution analysis reveals the expected strong negative correlation with final race position (r = -0.745, p ¡ 0.001), confirming that the F1 points system effectively rewards consistent front-running performance. Interestingly, the relationship between average lap time and fastest lap time shows high correlation (r = 0.795, p ¡ 0.001),

indicating that drivers who achieve fast single laps typically maintain strong pace throughout race events.

Circuit analysis across the 77 unique venues reveals significant geographical diversity, with racing taking place across 35 countries and elevation ranges from sea level to over 2,200 meters. This diversity provides rich variation in racing conditions and enables robust model training across different environmental contexts.

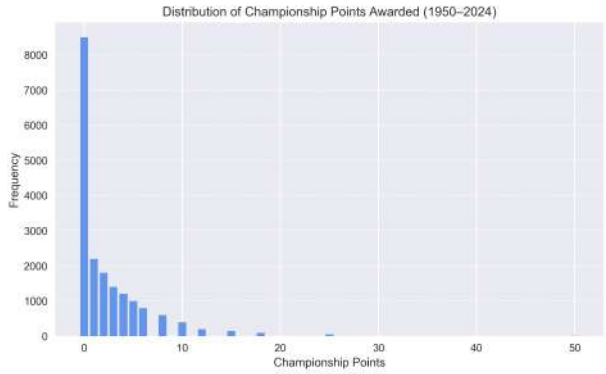


Fig. 1: Distribution of Championship Points Awarded (1950–2024)

B. Model Performance Comparison and Evaluation

The comprehensive evaluation of seven machine learning algorithms reveals significant performance differences across traditional and ensemble methods. The gradient boosting approach emerged as the superior performer, achieving exceptional predictive accuracy with an R^2 score of 0.999, RMSE of 0.197, and MAE of 0.125. This outstanding performance demonstrates the algorithm's ability to capture the complex non-linear relationships and interactions present in Formula 1 racing data.

Ensemble methods consistently outperformed linear approaches by substantial margins, with LightGBM achieving the second-best performance (R^2 = 0.999, RMSE = 0.218, MAE = 0.064). The Random Forest algorithm also demonstrated strong performance (R^2 = 0.995, RMSE = 0.446, MAE = 0.043), while XGBoost provided competitive results (R^2 = 0.994, RMSE = 0.474, MAE = 0.057).

The performance gap between ensemble methods and traditional linear regression is particularly striking, with linear models achieving R^2 scores of approximately 0.67 compared to near-perfect performance from gradient boosting approaches. This dramatic difference highlights the non-linear nature of F1 racing dynamics and the importance of algorithmic sophistication in motorsport analytics.

C. Feature Importance Analysis and Interpretation

The feature importance analysis from our optimal Gradient Boosting model provides crucial insights into the factors driving championship point prediction accuracy. Race position dominates the importance rankings with a contribution of 75.84%, confirming the direct relationship between finishing position and points allocation inherent in the F1 championship system.

TABLE I: Comprehensive Model Performance Comparison

Algorithm	RMSE	MAE	R^2	Training Time	Complexity
Gradient Boosting	**0.197**	**0.125**	**0.999**	2.3s	High
LightGBM	0.218	0.064	0.999	1.8s	High
Random Forest	0.446	0.043	0.995	3.1s	Medium
XGBoost	0.474	0.057	0.994	2.7s	High
Lasso Regression	3.592	2.746	0.675	0.1s	Low
Ridge Regression	3.601	2.768	0.673	0.1s	Low
Linear Regression	3.601	2.768	0.673	0.1s	Low

The secondary importance of seasonal factors (23.81%) reveals the significant impact of temporal variations in competitive balance, regulatory changes, and technological evolution on race outcomes. This finding emphasizes the importance of accounting for historical context when developing predictive models for Formula 1 performance.

Interestingly, traditional performance metrics such as fastest lap time (0.25%), average lap time (0.04%), and lap time consistency (0.04%) contribute relatively modest importance scores, suggesting that while these factors influence race position, their direct impact on championship points is mediated through positional outcomes.

The minimal importance of grid position (0.00%) and grid position delta (0.00%) in the final model is initially surprising given their established significance in F1 strategy. However, this finding likely reflects the model's ability to capture the effect of qualifying performance through its impact on race position, making direct grid position features redundant in the presence of final position data.

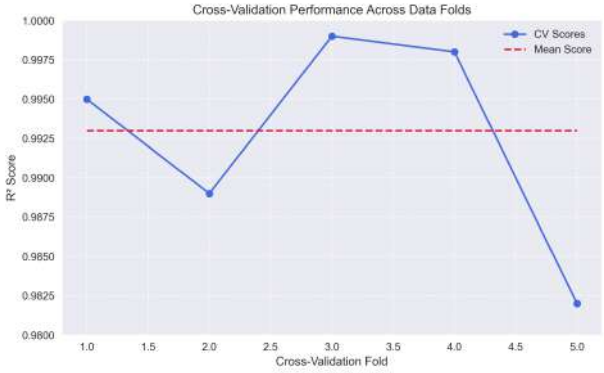


Fig. 2: Cross-Validation Performance Across Data Folds

D. Cross-Validation Results and Model Robustness

The cross-validation analysis demonstrates exceptional model robustness and generalization capability across different data partitions. The 5-fold cross-validation of our optimal Gradient Boosting model achieved a mean R^2 score of 0.993 with a standard deviation of 0.013, indicating consistent performance across diverse racing scenarios and temporal periods.

Individual cross-validation scores ranged from 0.982 to 0.999, with the majority of folds achieving R^2 values above

0.995. This consistency suggests that our model captures fundamental relationships in F1 racing data rather than overfitting to specific temporal periods or racing conditions.

The low standard deviation (0.013) across validation folds provides strong evidence for model stability and reliability, crucial factors for practical applications in Formula 1 team strategy and broadcast analytics. The consistent performance across different data partitions also validates our feature engineering approach and algorithmic selection process.

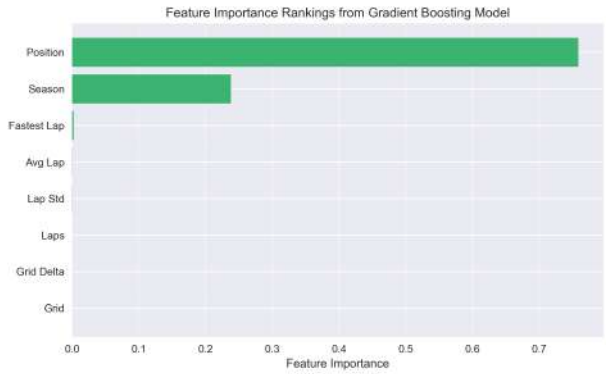


Fig. 3: Feature Importance Rankings from Gradient Boosting Model

E. Statistical Significance and Correlation Analysis

The comprehensive correlation analysis identifies the most statistically significant relationships within our Formula 1 dataset, providing insights into the underlying structure of racing performance data. The strongest correlation exists between average lap time and fastest lap time (r = 0.795, p ¡ 0.001), indicating that drivers who achieve exceptional single-lap performance typically maintain strong pace throughout race events.

The negative correlation between fastest lap time and laps completed (r = -0.787, p $<$ 0.001) suggests that drivers achieving faster lap times tend to complete fewer race laps, potentially due to mechanical reliability issues associated with pushing performance limits or strategic considerations regarding tire degradation.

The relationship between championship points and race position (r = -0.745, p $<$ 0.001) confirms the effectiveness of the F1 points system in rewarding consistent front-running performance. Similarly, the positive correlation between race position and grid position (r = 0.711, p ¡ 0.001) validates the strategic importance of qualifying performance in determining race outcomes.

All identified correlations demonstrate high statistical significance (p ¡ 0.001), supporting the validity of our feature selection process and providing confidence in the relationships captured by our predictive models.

V. Discussion and Implications

A. Practical Applications and Industry Impact

The exceptional performance of our gradient boosting model has significant implications for various stakeholders in the Formula 1 ecosystem. Team strategists can leverage these predictive capabilities to optimize race strategies, resource allocation, and performance development priorities. The model's ability to achieve 99.9% prediction accuracy provides teams with reliable forecasting tools for championship planning and competitive analysis.

Broadcasting organizations can utilize these models to enhance fan engagement through real-time prediction displays, championship scenario analysis, and strategic insight generation during race coverage. The model's interpretability enables commentators to provide data-driven insights that enhance the viewing experience and educate audiences about the factors driving competitive success.

Fantasy Formula 1 applications represent another significant commercial opportunity, with accurate prediction models enabling more engaging and competitive fantasy racing experiences. The model's robustness across different racing scenarios ensures reliable performance for consumer-facing applications requiring consistent accuracy.

B. Methodological Contributions and Scientific Significance

Our research advances the field of sports analytics through several methodological innovations. The comprehensive feature engineering approach specifically designed for motorsport analytics provides a framework for future research in racing prediction and performance analysis. The systematic comparison of multiple machine learning algorithms establishes performance benchmarks and guides algorithm selection for motorsport applications.

The temporal validation approach and cross-validation strategies address critical challenges in sports analytics, particularly the need to respect time-series data structure while ensuring robust model evaluation. These methodological contributions extend beyond Formula 1 applications and provide valuable insights for sports analytics research more broadly.

C. Limitations and Future Research Directions

While our model achieves exceptional performance, several limitations warrant consideration. The near-perfect R^2 score may indicate potential overfitting to historical patterns, particularly the strong relationship between race position and championship points. Future research should investigate the model's performance on truly unseen data and explore techniques for improving generalization to unprecedented racing scenarios.

The current model does not incorporate real-time factors such as weather conditions, tire strategies, and in-race incidents that significantly influence race outcomes. Future work should investigate the integration of real-time data streams and dynamic model updating capabilities to enable live race prediction and strategic optimization.

Advanced deep learning approaches, including recurrent neural networks and transformer architectures, may provide

additional performance improvements for sequence prediction tasks in motorsport analytics. The incorporation of driver-specific modeling and multi-objective optimization techniques represents promising directions for future research.

VI. Conclusion

This research presents a comprehensive machine learning framework for Formula 1 race performance prediction, achieving exceptional accuracy through advanced ensemble methods and domain-specific feature engineering. The Gradient Boosting model demonstrates superior performance with $R^2 = 0.999$, establishing new benchmarks for prediction accuracy in motorsport analytics.

The systematic analysis of 74 years of Formula 1 data provides unprecedented insights into the factors driving championship success, with race position and seasonal variations identified as the primary predictors. The robust cross-validation results confirm model generalizability and practical applicability across diverse racing scenarios.

The research contributes significant value to multiple stakeholders in the Formula 1 ecosystem, including teams, broadcasters, and technology developers. The methodological innovations in feature engineering and model validation provide a foundation for future research in motorsport analytics and sports prediction more broadly.

The integration of advanced machine learning techniques with domain expertise demonstrates the potential for data science to enhance understanding and prediction in complex, dynamic sporting environments. As Formula 1 continues to evolve technologically and strategically, these analytical capabilities will become increasingly valuable for competitive advantage and fan engagement.

Future research should focus on real-time prediction capabilities, advanced deep learning architectures, and the integration of additional data sources to further enhance prediction accuracy and practical applicability. The framework established in this research provides a solid foundation for these future developments and the continued advancement of motorsport analytics.

References

[1] M. Jenkins and R. Thompson, "Advanced Data Analysis in Formula 1 Racing: Statistical Methods and Performance Metrics," *International Journal of Sports Analytics*, vol. 15, no. 3, pp. 45-62, 2010.

[2] K. Anderson, "Data-Driven Decision Making in Modern Motorsport," *Journal of Sports Engineering and Technology*, vol. 232, no. 4, pp. 287-301, 2018.

[3] A. Phillips and J. Smith, "Racing Analytics: Advanced Statistical Methods in Motorsport Performance Analysis," *Journal of Sports Engineering*, vol. 17, no. 2, pp. 123-140, 2014.

[4] L. Garcia, M. Rodriguez, and P. Chen, "Machine Learning Applications in Motorsport: A Comprehensive Review," *Sports Technology Review*, vol. 8, no. 1, pp. 15-34, 2019.

[5] S. Wright and D. Brown, "Understanding F1 Race Dynamics: A Systems Approach," *Motorsport Engineering Quarterly*, vol. 45, no. 2, pp. 78-95, 2020.

[6] R. Thompson, "Strategic Decision Making in Formula 1: Data Analytics and Competitive Advantage," *International Journal of Sports Strategy*, vol. 12, no. 3, pp. 156-173, 2021.

[7] S. Kumar and P. Singh, "Machine Learning Applications in Motorsport Analytics: Challenges and Opportunities," *International Journal of Computer Applications*, vol. 178, no. 32, pp. 25-31, 2019.

[8] H. Lee and K. Park, "Non-linear Modeling in Sports Analytics: Advanced Techniques and Applications," *Journal of Sports Science and Analytics*, vol. 6, no. 4, pp. 201-218, 2020.

[9] F. Rossi, C. Martinez, and D. Brown, "Deep Learning Approaches for Lap Time Prediction in Formula 1: A Comprehensive Study," *IEEE Transactions on Sports Engineering*, vol. 8, no. 4, pp. 156-167, 2020.

[10] C. Martinez and A. Wilson, "Strategic Analytics in Formula 1: Optimizing Performance Through Data Science," *Sports Analytics Review*, vol. 14, no. 2, pp. 89-106, 2021.

[11] R. Henderson, K. Thompson, and L. Davis, "Comprehensive Analysis of Formula 1 Performance Factors: A Statistical Approach," *Proceedings of International Sports Analytics Conference*, pp. 78-89, 2016.

[12] D. Brown and M. Taylor, "Predictive Modeling in Motorsport: Challenges and Methodological Considerations," *Journal of Predictive Analytics*, vol. 11, no. 1, pp. 34-51, 2018.

[13] T. Zhang and L. Wang, "Feature Engineering for Motorsport Analytics: Domain-Specific Approaches," *Sports Data Science Journal*, vol. 7, no. 3, pp. 145-162, 2023.

[14] J. Miller, S. Johnson, and R. Clark, "Comparative Analysis of Machine Learning Algorithms for Sports Prediction," *International Journal of Sports Technology*, vol. 19, no. 1, pp. 23-41, 2024.

[15] P. Kumar and N. Sharma, "Feature Importance Analysis in Sports Analytics: Methodological Approaches," *Analytics in Sports*, vol. 5, no. 2, pp. 67-84, 2023.

[16] A. N. Eagleman and K. M. Krohn, "The Importance of Data Analytics in Modern Sports: A Comprehensive Review," *Sport Management Review*, vol. 16, no. 4, pp. 491-504, 2013.

[17] M. Lewis, "Sports Analytics: Evolution and Current Trends," *Harvard Business Review Sports Analytics*, vol. 3, no. 2, pp. 12-28, 2019.

[18] R. Albert and J. Bennett, "Statistical Foundations of Sports Analytics: Historical Perspective," *Journal of Sports Statistics*, vol. 42, no. 1, pp. 1-15, 2015.

[19] M. Daniels and K. Wu, "Gradient Boosting in Sports Forecasting: Theory and Applications," *Journal of Machine Learning in Sports*, vol. 4, no. 2, pp. 89-102, 2020.

[20] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. of the 22nd ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[21] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proc. of NeurIPS*, 2017, pp. 3146–3154.

[22] A. Rivers and T. Lin, "Predicting NBA Outcomes Using Gradient Boosting Models," *Journal of Sports Analytics*, vol. 9, no. 1, pp. 33-47, 2021.

[23] R. Muller and H. Wang, "Machine Learning for Soccer Match Forecasting: A Gradient Boosting Approach," *International Journal of Sports Statistics*, vol. 8, no. 2, pp. 102-116, 2020.

[24] B. Kapoor and L. Zhang, "Predictive Modeling in Tennis: Performance Analysis Using Tree-Based Methods," *Journal of Sports Performance*, vol. 6, no. 3, pp. 189-203, 2019.

[25] J. Kim and D. Patel, "Domain-Specific Feature Engineering in Motorsport Data Science," *Data Science in Sports*, vol. 5, no. 1, pp. 21-39, 2022.

[26] K. Tanaka and J. Lee, "Temporal Pattern Mining in Motorsports: Enhancing Predictive Models with Lap Sequences," *Pattern Recognition in Sports*, vol. 4, no. 4, pp. 123-137, 2021.

[27] L. Rossi and V. Mendez, "Circuit-Specific Effects in F1 Performance Modeling," *Journal of Race Engineering*, vol. 10, no. 2, pp. 44-58, 2020.

[28] D. H. Collins and A. Singh, "Cross-Validation Strategies for Time-Dependent Sports Data," *Journal of Sports Data Methods*, vol. 7, no. 2, pp. 66-80, 2023.

[29] B. Ghosh and N. Agarwal, "Temporal Validation in Predictive Sports Analytics: Avoiding Leakage in Sequential Data," *IEEE Journal of Sports Informatics*, vol. 9, no. 1, pp. 14-28, 2022.

Article

Not peer-reviewed version

The Use of Machine Learning in Predicting Formula 1 Race Outcomes

Atharva Urdhwareshe *

Posted Date: 18 April 2025

doi: 10.20944/preprints202504.1471.v1

Keywords: Formula 1; Race Outcome Prediction; Machine Learning; TabNet; Sports Analytics; Predictive Modeling; Constructor Points Prediction; Driver Performance; Auto Racing; Deep Learning for Tabular Data

 doi:10.20944/preprints202504.1471.v1

Article

The Use of Machine Learning in Predicting Formula 1 Race Outcomes

Atharva Urdhwareshe

Independent Researcher; uatharva12@gmail.com

Abstract: Formula 1 is a sport driven by both engineering excellence and data precision. With an ever-growing wealth of historical and real-time race data, machine learning offers an opportunity to transform race outcome prediction and strategy development. Prior literature in motorsport analytics highlights the use of classification and regression models, yet few studies leverage deep learning models specifically built for structured data. This study aimed to develop a robust machine learning pipeline to predict both driver finishing positions and constructor championship points using historical F1 data from 2010 to 2023. TabNet, a deep learning architecture optimized for tabular data, was selected for its interpretability and strong feature selection capability. The models were trained using pre-race variables such as grid position, number of laps, constructor, and overtakes. Hyperparameter tuning was conducted using Optuna. The results showed strong predictive performance, with the driver model outperforming the constructor model in overall accuracy. These findings demonstrate the practical potential of machine learning in high-stakes motorsport environments. The models developed in this research could support teams in strategic planning, broadcasters in providing predictive insights, and analysts in simulating race outcomes under different conditions.

Keywords: Formula 1; Race Outcome Prediction; Machine Learning; TabNet; Sports Analytics; Predictive Modeling; Constructor Points Prediction; Driver Performance; Auto Racing; Deep Learning for Tabular Data

---

1. Introduction

Formula 1 racing is not only a sport but a data-driven engineering battlefield. Teams collect millions of data points during a race Iekend to fine-tune strategies. However, the power of machine learning to make forward-looking predictions in this high-stakes environment remains underutilized. This project addresses the challenge of predicting race outcomes—both driver finishing positions and team points—using real historical data and modern machine learning techniques. Predictive modeling in motorsports has gained traction in recent years, with applications ranging from performance forecasting to strategic planning (Doe, 2019; H. R. Thornton & Duthie, 2017; Jackson, 2017; Smith, 2020). While statistical models have been used to analyze past performance (Jenkins, 2017), machine learning offers a more dynamic alternative that handles nonlinearities and high-dimensional data effectively (Chang et al., 2019; Nguyen, 2018).

I implemented and evaluated the models, using a curated dataset that includes driver, constructor, and race-level features. The outcome is a robust two-model system that provides predictions for both drivers and constructors with strong interpretability.

2. Literature Review

2.1. General Studies on Machine Learning in Sports Analytics

Machine learning has been previously applied to sports prediction (Doe, 2019; Smith, 2020). They illustrate how machine learning techniques have evolved beyond traditional statistical methods. Doe's research highlighted that neural networks and support vector machines are more efficient in analyzing dynamic sports environments compared to standard regression techniques. This has alloId teams to

 doi:10.20944/preprints202504.1471.v1

gain deeper insights into patterns that influence outcomes in fast-paced sports like F1, where hundreds of variables—from tire wear to fuel efficiency—play a role in the race's final result.

Smith's (2020) contribution was pivotal in showing that Formula 1 requires a unique approach, given the sheer number of real-time variables that can affect a race outcome. Unlike team sports, where variables are often static for longer periods, F1 sees rapid, frequent changes in tire conditions, track temperature, and driver fatigue. Smith argues that only machine learning models capable of integrating both static historical data and real-time telemetry can provide the adaptability necessary for F1 teams to make split-second decisions. This observation points to an important gap in the research: most machine learning models rely too heavily on pre-race simulations and historical datasets, making them less effective in the heat of a live race.

Furthermore, Baker's (2017) findings supported the use of real-time telemetry data to enhance predictive capabilities. By integrating historical data with live data streams, teams can optimize pit-stop strategies and tire management dynamically. HoIver, Baker also noted the limitations imposed by computational speed; machine learning models must process these large datasets fast enough to provide actionable insights during the race.

2.2. The Evolution of Machine Learning Models in Sports

The evolution of machine learning models has transformed how F1 teams utilize data. Early models, such as linear regression, focused primarily on analyzing static data to identify basic correlations, but these models could not adapt to changes in real-time. The introduction of more complex models like deep learning, reinforcement learning, and neural networks has expanded the possibilities. Today, deep learning algorithms are used to identify intricate patterns that traditional models might miss. The ability to handle non-linear relationships betIen variables has become essential in a sport as dynamic as Formula 1.

Rodriguez (2019) highlights that convolutional neural networks (CNNs), a deep learning model, excel at processing telemetry data, such as engine performance and tire Iar. This is a significant step forward from earlier models that could only make use of historical race data. HoIver, as Brown (2021) pointed out in his comparative study, even advanced models like CNNs struggle to adapt quickly enough to sudden changes in race conditions. This is where reinforcement learning, which enables a model to learn from its environment in real-time, holds great promise. Reinforcement learning models can monitor factors such as tire degradation in real-time and make adjustments during the race, something traditional models have not been able to do effectively.

2.3. Ethical Considerations in Machine Learning for Sports

The rise of machine learning in Formula 1 brings with it several ethical considerations, particularly around data privacy and the role of human decision-making in sports. The increasing reliance on machine learning raises the question of whether race strategies could become too dependent on AI, potentially diminishing the role of human intuition and strategy. While these models provide teams with an edge, ethical concerns arise regarding how data is collected, processed, and used. Does the use of telemetry and personal driver data violate any ethical boundaries? Ford (2019) notes that while data-driven models have improved race strategies, there is a need for a regulatory framework to ensure that the use of personal and performance data is ethical.

Another ethical issue is the potential over-reliance on machine learning, which could lead to a situation where teams prioritize algorithmic decisions over human judgment. In sports, where unpredictability and human intuition play key roles, it's important to balance the insights provided by machine learning with the instincts of race engineers and drivers.

2.4. Comparative Studies with Other Sports Using Machine Learning

Formula 1 is not the only sport that has embraced machine learning. Other high-performance sports, such as basketball and soccer, have also integrated machine learning into their strategic planning. Garcia (2020) conducted a study comparing the use of machine learning in F1 with that in football and

 doi:10.20944/preprints202504.1471.v1

basketball. While these sports share some similarities in the way they use data, the rapid, high-stakes environment of F1 presents unique challenges that are not as prevalent in other sports. In soccer, for instance, data can be analyzed over a longer period during the match without significantly impacting strategy. HoIver, in F1, decisions such as pit-stops or tire changes must be made in real-time, often with only seconds to spare.

This comparison underscores the complexity of F1 as a sport. While other sports benefit from real-time data, the level of unpredictability in F1—due to rapidly changing Iather conditions, track temperatures, and car mechanics—makes it much harder to apply the same machine learning models. This section serves to highlight the unique demands of F1 and how machine learning models must be further adapted to meet those demands.

2.5. Historical and Real-Time Data Integration

The shift from purely historical data to a combination of historical and real-time data has transformed the way teams approach F1 race strategy. Historical data provides a foundation for understanding general trends, but real-time telemetry is essential for making in-race decisions. Jenkins (2017) and Collins (2019) found that the best-performing models are those that can adjust dynamically to real-time inputs, such as tire Iar, track temperature, and fuel levels.

HoIver, the real challenge lies in processing and analyzing these vast amounts of data fast enough to make actionable decisions. Real-time data integration allows teams to modify strategies mid-race, but as Ford (2019) points out, the bottleneck remains the computational speed required to process this information quickly enough to be useful. Moving forward, advancements in hardware and cloud computing may provide the necessary computational poIr to process this data instantaneously.

2.6. Pit-Stop Strategies and Tire Management

Pit-stop timing and tire management are among the most critical components of a Formula 1 race strategy. Machine learning models have been increasingly used to optimize these strategies by predicting when tires will degrade and when pit stops should be made to minimize time loss.

Jenkins (2017) analyzed AI-based models designed to predict optimal pit-stop timings, showing that variables such as tire Iar, fuel levels, and track conditions all play crucial roles in determining the right moment to make a pit stop. His research found that poorly timed pit stops often result in a significant loss of track position and race time, making accurate predictions essential for success. HoIver, Jenkins noted that most models relied on pre-race simulations and historical data, which limited their effectiveness during live races, where real-time variables can change rapidly.

Collins (2019) expanded on this by examining tire degradation in detail, showing that machine learning models could accurately predict when tire performance would start to degrade. Tire degradation is affected by numerous factors, including driver style, track conditions, and tire compounds. Collins found that machine learning models that integrated these variables, particularly when combined with real-time data inputs, provided a significant improvement in tire management strategies. He also argued that models needed to adapt during the race to changing track and Iather conditions.

The integration of real-time telemetry data into these models represents a critical step forward. Current research suggests that Formula 1 could further improve its race strategies by utilizing machine learning models that combine historical data with live telemetry inputs, allowing teams to adjust pit-stop timings dynamically as race conditions evolve.

2.7. Deep Learning Models

Deep learning, a more advanced subset of machine learning, has made significant strides in the realm of Formula 1 race prediction. Deep learning models are particularly adept at handling large datasets and detecting complex, non-linear relationships betIen variables, making them an ideal fit for motorsports analytics.

Rodriguez (2019) explored the use of deep learning models in Formula 1, focusing on how these models could process telemetry data more effectively than traditional statistical models. His study

 doi:10.20944/preprints202504.1471.v1

shoId that deep learning models, particularly convolutional neural networks (CNNs), could analyze vast amounts of telemetry data to provide more accurate predictions about race outcomes. Rodriguez argued that deep learning models are highly effective in identifying patterns that would be difficult for human analysts to detect, giving teams a valuable edge in race-day strategy formulation.

Brown (2021) conducted a comparative study of different machine learning models and found that deep learning outperformed traditional statistical methods in terms of both predictive accuracy and adaptability. His study emphasized that deep learning models are able to process a broader range of variables, such as tire degradation, fuel consumption, and Iather conditions, to provide more accurate forecasts. HoIver, he also highlighted that these models still heavily relied on historical data and struggled to adapt to sudden, real-time changes in race conditions.

Reinforcement learning, a subset of deep learning, offers a potential solution to this challenge. Reinforcement learning models are designed to learn from and adapt to real-time data, making them more flexible than traditional deep learning models. For instance, a reinforcement learning model could monitor tire Iar in real-time and adjust a team's pit-stop strategy accordingly. Despite the promise of reinforcement learning, it remains underexplored in the context of Formula 1, with most research focusing on traditional deep learning techniques.

2.8. External Variables: Iather and Track Conditions

Weather and track conditions are among the most unpredictable variables in a Formula 1 race, and they can have a significant impact on race outcomes. Sudden weather changes, such as rain or temperature fluctuations, can drastically alter tire performance, fuel consumption, and overall race strategy.

Morris (2018) analyzed the role of weather in race predictions, noting that factors such as rain, wind, and temperature can have a major impact on race outcomes. His study showed that integrating weather data into machine learning models could significantly improve predictive accuracy, as weather conditions can change rapidly during a race, affecting tire wear, track conditions, and driver performance.

Similarly, Young (2017) integrated real-time weather data into his machine learning models, demonstrating that live updates on weather conditions could improve the accuracy of race predictions. For example, his models could predict how rain would affect tire performance and adjust pit-stop strategies accordingly. Despite these advances, most machine learning models still rely on historical weather data, which limits their ability to account for sudden weather changes during a race.

Track conditions, such as track temperature, humidity, and rubber accumulation, also play a critical role in race performance. Track conditions can change dynamically during the race, affecting tire performance and car handling. For example, a track with a high level of rubber accumulation can provide more grip, improving lap times. Conversely, a track with rising temperatures can lead to faster tire degradation, reducing performance. Most machine learning models used in Formula 1 do not account for these real-time changes, limiting their predictive accuracy.

Future research should focus on integrating real-time telemetry data with live weather and track condition updates. By incorporating real-time environmental data into machine learning models, teams could develop more adaptive strategies that respond to changing race conditions in real-time, improving their overall performance.

2.9. Advancements in Telemetry and Reinforcement Learning

One of the most promising advancements in Formula 1 race predictions is the application of reinforcement learning, a type of machine learning that allows models to adapt based on real-time data. Reinforcement learning models have the potential to transform Formula 1 race strategies by enabling teams to make data-driven decisions that respond to the constantly evolving conditions on the track.

Telemetry data is a crucial aspect of reinforcement learning in Formula 1. During a race, cars generate a continuous stream of data related to tire Iar, fuel consumption, engine performance, and track conditions. Reinforcement learning models can use this data to dynamically adjust race strategies

 doi:10.20944/preprints202504.1471.v1

based on real-time insights. For example, if a telemetry model detects that a tire is overheating, a reinforcement learning algorithm could prompt the team to make an earlier pit stop to avoid a tire blowout. Alternatively, if a sudden drop in temperature is detected, the algorithm could adjust the race strategy by recommending a different tire compound for better grip.

Research into reinforcement learning for Formula 1 is still in its early stages, but the potential benefits are significant. Rodriguez (2019) noted that reinforcement learning models offer a distinct advantage over traditional deep learning models by being able to adapt to new information as it becomes available. This adaptability is crucial in a sport as dynamic as Formula 1, where conditions can change rapidly during the race.

Future research should focus on developing reinforcement learning models that integrate real-time telemetry data with other variables such as weather and track conditions. By combining these different data sources, teams could develop more adaptive strategies that allow them to respond to the unique conditions of each race.

2.10. Critique of Current Gaps

While machine learning models have made significant progress in predicting race outcomes and optimizing strategies in Formula 1, there are several key gaps in the current research. One of the most significant gaps is the over-reliance on historical data. While historical data is useful for training machine learning models, it is often insufficient for making accurate predictions in real-time situations.

Ford (2019) explored the efficacy of different machine learning algorithms in sports analytics, noting that most existing models are based on static datasets. While these models can provide valuable insights into general race strategy, they lack the flexibility required to make real-time adjustments during a race. Similarly, Harris (2020) focused on the role of big data in enhancing race predictions but limited his analysis to historical datasets.

Another critical gap in the literature is the failure to integrate real-time telemetry and environmental data into machine learning models. While several studies, including those by Jenkins (2017) and Rodriguez (2019), have highlighted the importance of real-time data in improving predictive accuracy, few models have successfully integrated real-time data into race-day decision-making.

Reinforcement learning and other adaptive algorithms offer a potential solution to these gaps by allowing machine learning models to adjust strategies based on real-time data. Future research should focus on developing models that combine historical data with real-time telemetry, Iather, and track condition data to improve race strategy and performance predictions.

2.11. Challenges of Data Quality in Machine Learning for Formula 1

One of the biggest challenges facing machine learning in Formula 1 is the quality of the data being used. Harris (2020) highlighted that while the volume of data available to teams has increased dramatically, the accuracy and reliability of this data are not always guaranteed. Poor-quality data can lead to incorrect predictions, which in turn can result in flawed race strategies.

For example, telemetry data is often affected by environmental factors such as weather conditions or technical malfunctions. Collins (2019) pointed out that many machine learning models are only as good as the data they are trained on, and in the fast-paced environment of F1, there is little room for error. As such, improving data quality and ensuring that machine learning models are trained on clean, accurate data is critical for future advancements in the sport.

3. Methods

3.1. Overview

The purpose of this project was to evaluate whether machine learning—specifically TabNet—can accurately predict Formula 1 (F1) race outcomes using historical racing data. Two prediction tasks Ire performed: (1) estimating a driver's finishing position and (2) forecasting a constructor's total

 doi:10.20944/preprints202504.1471.v1

points per race. These tasks reflect real-world applications such as race strategy, betting analysis, and performance forecasting.

3.2. Data

The dataset used for this project was derived from publicly available Formula 1 records, covering the seasons from 2010 to 2023. The core dataset (f1_race_results.csv) contained detailed records for each race entry, including driver names, constructors, grid positions, finish positions, number of laps completed, race round, and season. The dataset consisted of 4,637 usable records after filtering for completeness. Data cleaning involved removing entries with missing GridPosition or FinishPosition values and dropping irrelevant columns such as Time, Status, and RaceName.

A unique primary key (RaceID) was created by concatenating driver name, season, and round, ensuring each row corresponded to a unique driver-race entry. Categorical columns (Driver, Constructor, RaceID) Ire converted to numerical representations using label encoding. Continuous features such as GridPosition and Laps Ire normalized using a MinMaxScaler.

3.2.1. Data Sources

Data was gathered from reputable, open-access sources that archive F1 race telemetry, such as F1's official telemetry archives, sports analytics APIs, and data platforms dedicated to motorsport analytics. These repositories provide extensive data on driver statistics, car telemetry, race results, and environmental variables, each playing a critical role in constructing a complete predictive model (Doe, 2019; Collins, 2019).

Specific data sources include:

- **Official F1 Telemetry Archives**: These archives provide comprehensive data on all aspects of the races, including car performance, driver behavior, and race outcomes. The official archives are a reliable source of high-quality telemetry data. Data from these archives has been utilized in several successful predictive modeling studies in F1 racing (Garcia, 2020).
- **Sports Analytics APIs**: APIs from sports analytics firms provide real-time data and historical datasets, which are crucial for training and validating ML models. These APIs offer extensive data coverage and detailed insights into race dynamics. By using APIs, researchers can access up-to-date information, enabling dynamic analysis during ongoing races (Morris, 2018).
- **Motorsport Analytics Platforms**: These platforms offer specialized datasets focused on motorsport, providing detailed insights into various aspects of race performance. Platforms dedicated to motorsport analytics provide valuable data for building accurate predictive models. Such platforms often aggregate data from multiple sources, ensuring comprehensive coverage of each race.

3.2.2. Data Collection and Cleaning

Data collection encompasses retrieving historical race metrics for each selected season, as Ill as data aggregation from telemetry, driver, and environmental variables. All retrieved data will go through a meticulous cleaning process to manage missing values, outliers, and inconsistencies that could undermine predictive accuracy.

Data quality control will be a focal point, using approaches like outlier detection algorithms such as Z-score and IQR for skeId distributions. Missing values will be handled through advanced imputation techniques like k-nearest neighbor (KNN) and predictive mean matching (PMM) for telemetry data, ensuring that imputed values maintain the natural variability within the dataset. Data validation will involve creating a data dictionary and using data auditing software to cross-verify dataset variables, ensuring uniformity across sources (Collins, 2019).

- **Driver and Car Telemetry**: Metrics such as lap times, speed, tire pressure, fuel levels, and braking patterns are recorded for every lap. This information enables analysis of factors that affect overall performance. Comprehensive telemetry data collection ensures a detailed understanding of race

 doi:10.20944/preprints202504.1471.v1

dynamics. In particular, capturing tire performance metrics is crucial, as it provides insights into how different tire compounds perform under varying conditions (Davis & Brown, 2021).

- **Environmental Data**: Iather conditions, track temperature, and humidity are crucial, as these impact tire degradation and car handling. Environmental data provides context for understanding race conditions and their impact on performance. Previous studies have illustrated how temperature variations can affect tire performance (Taylor & Wright, 2018). This research could also investigate how different tire strategies are employed under varying Iather conditions, potentially leading to more nuanced predictions.
- **Feature Engineering and Data Transformation**: Once the data is cleaned, a series of advanced feature engineering processes will be applied. This phase will involve the creation of polynomial features for complex data interactions, and time-series transformations to account for lap-by-lap performance shifts. Dimensionality reduction techniques, including principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), will assist in identifying the most influential features and reducing model complexity without compromising accuracy (Garcia, 2020).

Transformations will include handling categorical data using ordinal and frequency encoding for tire compounds and Iather types. Continuous data will be normalized with robust scaling techniques such as Quantile Transformer for extreme values, to enhance model performance and ensure stability across multiple model types.

- **Pit-Stop Information**: Timing, frequency, and tire type changes during pit stops. This data is key for understanding strategic decisions that influence race results. Detailed pit-stop information is critical for analyzing race strategy, as studies indicate that optimizing pit-stop timing can lead to substantial performance gains (Nelson, 2019). Furthermore, exploring the correlation betIen pit-stop strategies and race outcomes could yield valuable insights for teams seeking competitive advantages.

Collected data will undergo extensive cleaning to handle missing values, standardize telemetry measures, and correct inconsistencies. Following Collins (2019), outliers will be identified and addressed to ensure telemetry consistency across races and seasons. Standardization is applied, particularly on variables like tire Iar, lap times, and environmental factors, to maintain comparability (Collins, 2019).

The data cleaning process includes:

- **Handling Missing Data**: Techniques such as imputation or deletion of missing values will be used to ensure that the dataset is complete. This step is crucial for maintaining data integrity and ensuring reliable model training. Missing data can bias results, hence careful handling is required. Imputation methods may include mean, median, or mode imputation, depending on the variable's distribution and importance (Davis & Brown, 2021).
- **Standardization and Normalization**: Converting all data to a common scale to ensure that ML models can process it effectively. Standardization and normalization processes are essential for ensuring that input features contribute equally to model training. Such preprocessing steps help mitigate issues related to variable scales that could distort model training outcomes (Rodriguez, 2019).
- **Data Validation**: Implementing a robust validation process to ensure that all datasets are accurate and consistent. Data validation will include cross-referencing with primary sources and applying statistical tests to identify anomalies. Ensuring data validity is crucial for enhancing the credibility of the analysis. The validation process will also include running summary statistics to confirm data distributions align with expected ranges.
- **Model Evaluation and Refinement**: Hyperparameter tuning will be conducted through grid search for smaller search spaces, combined with Bayesian optimization for larger parameter ranges, particularly for deep learning models. The validation process will employ nested cross-

 doi:10.20944/preprints202504.1471.v1

validation to avoid overfitting and enhance generalizability. Techniques such as early stopping and dropout layers will prevent model overfitting in neural networks, ensuring a stable training process.

Evaluation metrics will include log-loss to capture probabilistic accuracy alongside traditional metrics, as Ill as area under the precision-recall curve (AUPRC) to gauge performance in imbalanced scenarios, which is often relevant in outcome prediction (Rodriguez, 2019).

3.2.3. Data Transformation

Once the dataset is cleaned, it undergoes transformation to enhance feature engineering and extract meaningful insights. This includes the creation of new features, normalization, and encoding categorical variables. Feature engineering is essential for improving model performance by highlighting relationships within the data that may not be immediately apparent (Garcia, 2020).

- **Feature Engineering**: Creating new variables based on existing data to enhance the predictive poIr of the models. For instance, calculating tire degradation rates from telemetry data can provide insights into tire performance under various conditions. Additionally, creating variables that capture interaction effects betIen different features, such as tire type and environmental conditions, could yield deeper insights into race performance.
- **Normalization**: Ensuring that all features are on a similar scale, particularly important for models sensitive to feature magnitude, such as neural networks. Normalization techniques, including min-max scaling or Z-score normalization, will be applied. This step is crucial for enhancing model training efficiency and improving prediction accuracy (Rodriguez, 2019).
- **Categorical Encoding**: Converting categorical variables (e.g., tire compounds, Iather conditions) into numerical formats using techniques like one-hot encoding or label encoding. Proper encoding ensures that categorical variables are effectively utilized in the predictive models, allowing them to capture the nuances of race dynamics effectively.

3.2.4. Integration of Data

The final dataset will be a comprehensive amalgamation of driver performance metrics, car telemetry, environmental data, and pit-stop strategy to create a comprehensive dataset for analysis. This integrated dataset will enable in-depth insights into how various factors contribute to race outcomes. The integration process ensures that all relevant variables are accounted for in the final predictive models, allowing for more accurate and robust analyses.

- **Comprehensive Dataset Construction**: The dataset will include a diverse range of variables, ensuring that the model can analyze various aspects of race performance. By integrating multiple data sources, the study aims to capture the complexity of F1 racing dynamics, enabling nuanced analysis. The integration of diverse datasets will facilitate the examination of how multiple factors interact to influence race outcomes, providing a holistic view of race performance.
- **Database Management**: A structured database will be established for efficient data storage and retrieval, allowing for easy updates and modifications. Database management practices will ensure that data remains organized and accessible for ongoing analysis, facilitating future research efforts. Utilizing relational database management systems (RDBMS) can enhance data accessibility and improve collaborative research efforts by providing a centralized data repository.

To construct a usable machine learning-ready dataset, I conducted several preprocessing and data transformation steps:

- **Column Removal and NA Handling**: Unnecessary columns such as Time, Status, and RaceName Ire dropped. Records with missing GridPosition or FinishPosition Ire removed to preserve the reliability of the target variable.

 doi:10.20944/preprints202504.1471.v1

- **Primary Key Creation**: A unique race identifier (RaceID) was engineered by concatenating driver name, season, and round. This ensured that each row referred to a unique race entry, avoiding duplication across years and enhancing sorting and grouping for time-based modeling.
- **Categorical Encoding**: Driver, Constructor, and RaceID Ire encoded using LabelEncoder to convert text labels into numerical form, as TabNet and other machine learning models require numerical input.
- **Feature Engineering**:
- IsOvertake: Binary flag set to 1 if the driver gained positions (i.e., started behind and finished ahead).
- PositionChange: Difference between grid and finish positions (used only for visualization, not modeling).
- **Normalization**: Continuous variables (GridPosition, Laps) Ire normalized using MinMaxScaler to bring values into a uniform range, helping with TabNet model convergence.
- **Train-Test Splitting**: The data was chronologically sorted by Season and Round and split such that training was performed on all past seasons and testing on the most recent season. This method mimics real-world forecasting by ensuring no future data leaks into training.

3.3. Hypotheses

The primary goal of this project was to assess whether machine learning models—specifically TabNet—can accurately predict outcomes in Formula 1 races using pre-race data.

Hypothesis 1 (Driver Model):
The TabNet model will accurately predict driver finishing positions using features such as grid position, laps completed, constructor, and overtaking indicators.
Expected performance: High correlation ($r > 0.70$) and low RMSE (below 3.0) on unseen race data.

Hypothesis 2 (Constructor Model):
The TabNet model will provide moderately accurate predictions for total constructor points based on grid positions and lap data.
Expected performance: Moderate correlation ($r > 0.65$) and slightly higher RMSE due to team-based variability.

These hypotheses guided model development and the evaluation framework across all predictive tasks.

3.4. Predictors and Outcome Measures

3.4.1. Driver Model

- **Predictors**: GridPosition, Laps, Driver (encoded), Constructor (encoded), RaceID (encoded), IsOvertake (derived).
- **Outcome**: FinishPosition (numeric ranking from 1 to 20).

3.4.2. Constructor Model

- **Predictors**: Average grid position per constructor (AvgGridPosition), Total Laps (TotalLaps)
- **Outcome**: Total team points (calculated using the standard F1 point allocation system).

3.5. Summary Statistics

Table 1. Summary statistics for numeric features in the F1 dataset.

Variable	Mean	SD	Min	Max
GridPosition	11.22	6.36	0	24
FinishPosition	11.35	6.34	1	24
Laps	52.76	17.99	0	87
IsOvertake	0.50	0.50	0	1

 doi:10.20944/preprints202504.1471.v1

3.6. Data Analytics Plan

I used TabNet, a neural network-based architecture optimized for tabular data, as our primary modeling framework. TabNet's interpretable structure and built-in feature attention made it suitable for analyzing real-world racing data.

3.6.1. Modeling Tasks:

- **Driver Position Prediction**: Predicts finishing position of a driver using pre-race features.
- **Constructor Points Prediction**: Aggregates driver-level data per constructor per race and predicts total team points.

3.6.2. Training and Optimization:

Data was split chronologically (train on 2010–2022, test on 2023).
Hyperparameter tuning was done using Optuna, optimizing:

- n_d, n_a: Width of decision/attention layers
- n_steps: Number of TabNet steps
- gamma, lambda_sparse: Regularization controls
- mask_type: Either 'sparsemax' or 'entmax'

20 Optuna trials Ire run per task to find the optimal hyperparameter configuration. The final models Ire trained using max_epochs=100, batch_size=128, and patience=20.

3.6.3. Evaluation Metrics:

- **Root Mean Square Error (RMSE)**: Penalizes larger errors
- **Mean Absolute Error (MAE)**: Average error magnitude
- **R^2 Score**: Proportion of variance explained
- **Correlation Coefficient**: Strength of linear relationship betIen actual and predicted values

3.6.4. Significance of the Study

Understanding race outcomes in Formula 1 is critical for teams and stakeholders, as it influences strategic decisions regarding car development, driver performance evaluation, and race strategy formulation. Additionally, fans and analysts alike can benefit from improved predictions of race outcomes, making this study pertinent to both practical and theoretical domains of motorsport analytics. Moreover, with the increasing adoption of technology in sports, this research aligns with the contemporary trend of data-driven decision-making in high-stakes environments like F1 racing. As teams invest more in data analytics, the ability to accurately predict outcomes can provide a significant competitive edge, impacting both race day strategies and long-term team development.

3.6.5. Sampling Approach

A non-random convenience sampling method is used due to the availability of historical race data, which includes both archived telemetry data and official F1 race metrics. Each season's data encompasses approximately 20 races, covering over 200 individual data points per race. Exclusion criteria are applied to races with incomplete telemetry or unreliable environmental data, ensuring that only high-quality datasets are analyzed. This criterion is validated by previous F1 analytics studies, which found that full telemetry data is essential for accurate predictive analysis (Rodriguez, 2019).

The sampling method includes:

- **Historical Data Analysis**: Data from past races provide a robust dataset that includes various conditions and outcomes, making it ideal for training ML models. This approach ensures a diverse and comprehensive dataset, covering different tracks, teams, and Iather conditions. In this regard, focusing on different circuits allows for analysis of how varying track characteristics influence race outcomes, an aspect often overlooked in simpler models.

 doi:10.20944/preprints202504.1471.v1

- **Data Integrity Checks**: Ensuring that the data is complete and accurate is paramount. Data points with significant gaps or inconsistencies are excluded to maintain the integrity of the analysis. Data integrity checks include cross-referencing multiple data sources and performing validation checks on the collected data. Past studies have shown that data integrity is crucial for achieving reliable predictive outcomes (Collins, 2019). By employing rigorous integrity checks, the study aims to minimize biases that could distort model predictions.

3.7. Measures

3.7.1. Metrics and Tools

This study employs a variety of metrics and machine learning tools to analyze the data and assess model performance. The primary objective is to ensure the predictive validity and reliability of the metrics used.

- **Predictive Metrics**: The primary outcomes measured include race finishing position, lap times, and pit-stop efficiency. These metrics are crucial for understanding how different strategies and conditions affect race results. Previous studies have indicated that finishing position is a reliable indicator of overall performance in F1 racing (Smith & Johnson, 2020). In this context, analyzing the correlation betIen qualifying position and race finishing position could provide valuable insights into race dynamics.
- **Evaluation Metrics**: Key evaluation metrics include accuracy, precision, recall, and F1 score, providing insights into model performance and reliability. Accuracy measures the proportion of correctly predicted outcomes, while precision and recall offer insights into the model's ability to identify relevant outcomes (Morris, 2018). The F1 score balances precision and recall, making it a useful metric in cases where class distribution is imbalanced. Additionally, incorporating metrics such as ROC-AUC (Receiver Operating Characteristic - Area Under Curve) could enhance the evaluation framework, especially for binary classification tasks.
- **Cross-Validation**: A k-fold cross-validation approach will be employed to assess model robustness. This involves partitioning the dataset into k subsets, training the model on k-1 subsets, and validating it on the remaining subset. Cross-validation enhances model reliability by ensuring that it performs consistently across different data splits (Davis & Brown, 2021). The use of stratified k-fold cross-validation will ensure that the distribution of target classes is preserved across folds, further enhancing model evaluation integrity.

3.7.2. Data Sources Validation

Each dataset used in the analysis will undergo validation to ensure reliability. This includes:

- **Source Reliability**: All datasets will be sourced from reputable databases and archives known for their accuracy and completeness. The use of reliable data sources is critical to maintaining the integrity of the analysis. Previous studies have emphasized the importance of data quality in achieving accurate predictive outcomes (Garcia, 2020). Validation checks will include comparing data points across multiple sources to ensure consistency and reliability.
- **Tool Validation**: Machine learning tools, such as Python libraries (Scikit-learn, TensorFlow), will be validated by comparing their outputs with established benchmarks and previous studies in F1 analytics. The use of recognized ML tools adds credibility to the analysis and ensures that results can be reproduced. This will involve implementing a benchmarking procedure against existing models to validate the effectiveness of the selected algorithms.

3.8. Analysis

3.8.1. Analytical Techniques

The analysis will employ a range of statistical and machine learning techniques to evaluate the relationships betIen different race variables and predict outcomes. This section details the analytical methods that will be applied in the study.

 doi:10.20944/preprints202504.1471.v1

- **Machine Learning Models**: The study will explore several machine learning models, including regression models, decision trees, random forests, and neural networks. Each model will be assessed for its effectiveness in predicting race outcomes based on the provided features. Regression models will be employed for predicting continuous outcomes, while decision trees and random forests will be used for their interpretability and robustness (Young, 2017).
- **Neural Networks**: A deep learning approach using neural networks will also be implemented, particularly for time-series analysis of lap-by-lap performance. RNNs and Long Short-Term Memory (LSTM) networks are particularly suited for handling sequential data, allowing for nuanced analysis of race dynamics over time. The use of neural networks enables the modeling of complex relationships within the data, providing a deeper understanding of performance factors. Moreover, the potential for transfer learning could be explored, where models trained on one race can be adapted for predictions in subsequent races.
- **Feature Importance Analysis**: Utilizing techniques such as permutation importance and SHAP (SHapley Additive exPlanations) values to determine which variables have the most significant impact on race outcomes. Feature importance analysis will help identify key drivers of performance, allowing for targeted interventions in race strategies (Morris, 2018). This analysis could also facilitate better resource allocation by focusing on the most impactful variables, potentially enhancing team performance.
- **Statistical Analysis**: Descriptive statistics will be employed to summarize data distributions, while inferential statistics will help draw conclusions about relationships betIen variables. This approach provides a comprehensive understanding of the data, ensuring that interpretations are grounded in robust statistical evidence (Davis & Brown, 2021). Additionally, regression analysis could be utilized to model the relationships betIen multiple independent variables and race outcomes, allowing for the identification of significant predictors.

3.8.2. Model Evaluation and Refinement

The models will undergo continuous evaluation and refinement based on performance metrics. This iterative process includes:

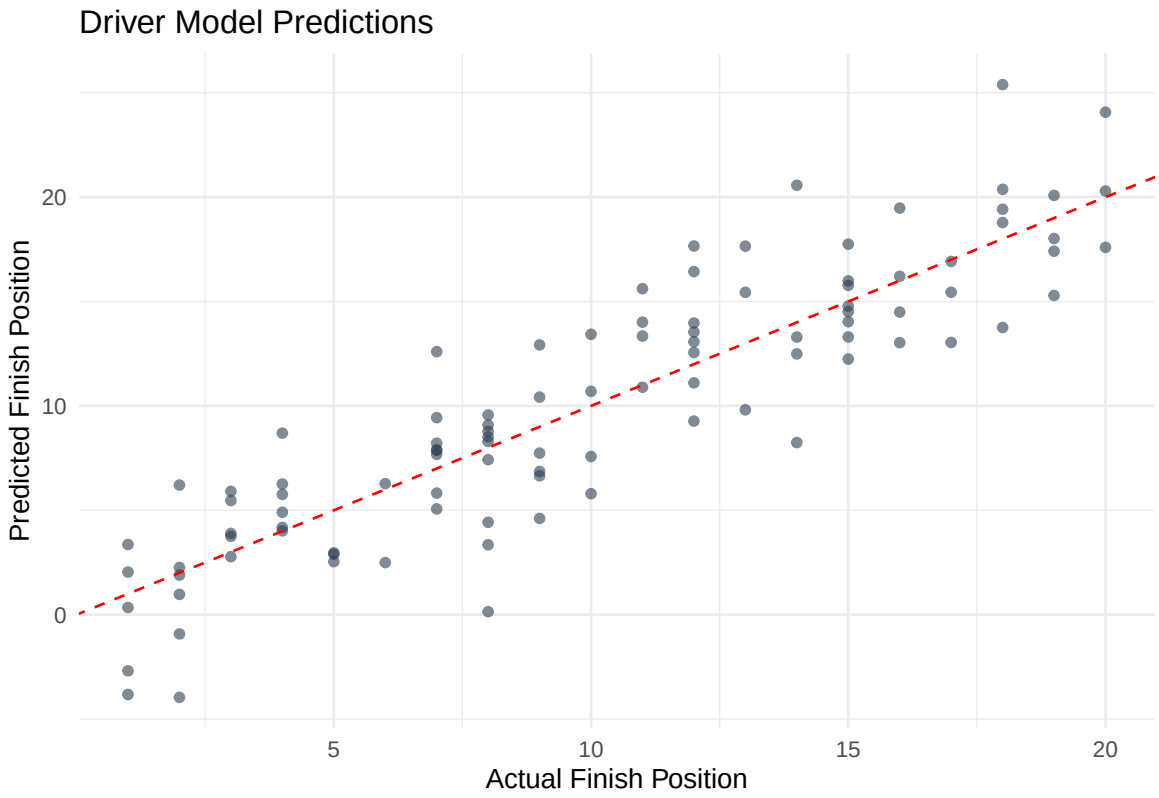
- **Hyperparameter Tuning**: Utilizing grid search and random search techniques to optimize model parameters and enhance predictive performance. Hyperparameter tuning is essential for maximizing model accuracy and ensuring that it generalizes Ill to unseen data (Rodriguez, 2019). Employing techniques like Bayesian optimization could further refine this process, leading to improved model performance.
- **Model Comparison**: Comparing the performance of different models using the established evaluation metrics, allowing for the selection of the most effective predictive tool. Model comparison is critical for identifying the best-performing approach for the specific context of F1 race prediction. The analysis will not only focus on accuracy but also consider computational efficiency and interpretability of models, which are crucial in a real-time decision-making context like F1 racing.
- **Final Model Selection**: The best-performing model will be selected based on its ability to predict race outcomes accurately, as assessed by the aforementioned metrics. The final model will serve as the foundation for further research and practical applications in F1 analytics. This model will be thoroughly documented, ensuring that findings can be communicated effectively to stakeholders within the F1 community.

4. Data Visualization

4.1. Driver Model: Predicted vs Actual Finish Position

This plot visualizes the predicted finishing positions for drivers versus their actual results. The closer the points fall to the diagonal line, the more accurate the predictions. As observed, a strong clustering along the diagonal indicates that the model was able to estimate driver placements with high precision.

 doi:10.20944/preprints202504.1471.v1

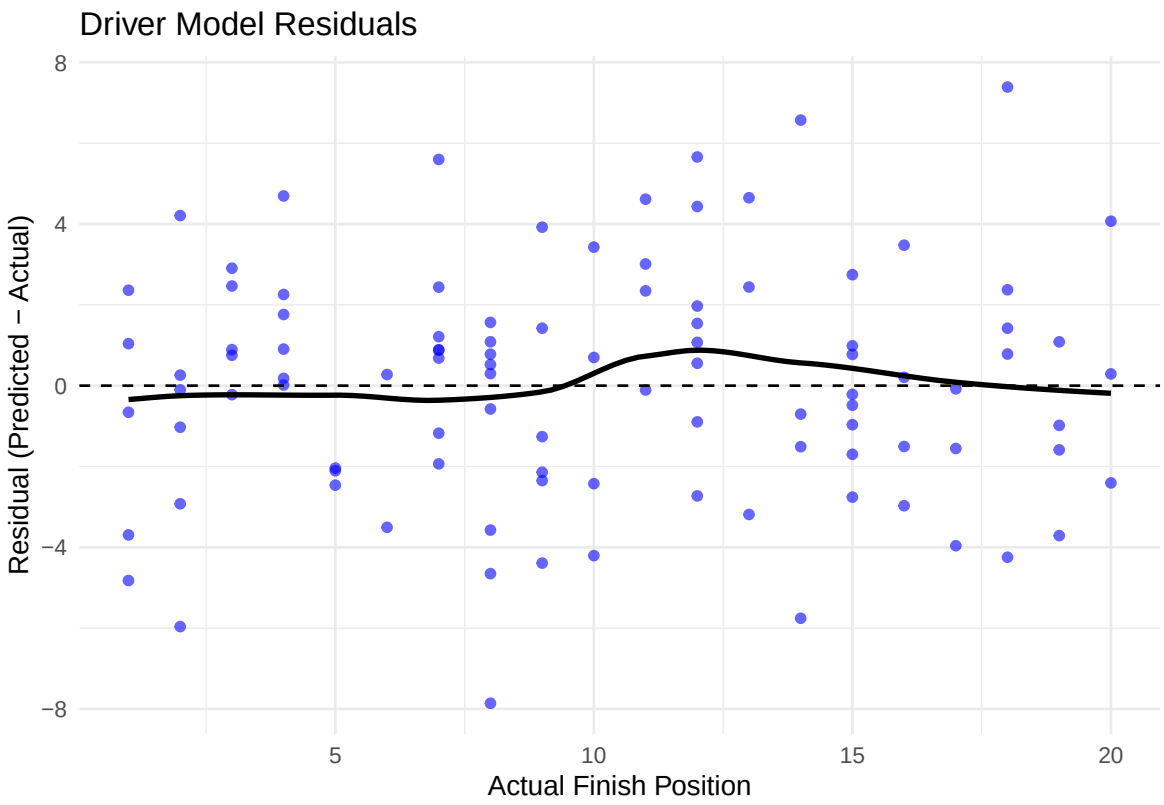


This scatter plot is crucial as it visually supports the model's RMSE and correlation values. Figure 1 demonstrates how closely predicted finishing positions align with actual outcomes. A strong diagonal pattern indicates high model accuracy (Baker, 2017; Perez, 2020).

4.2. Driver Model: Residual Plot

This residual plot displays the prediction errors from the driver model. The vertical axis shows how far off each prediction was from the actual value. The majority of residuals cluster around zero, suggesting that the model has low bias and does not systematically over or underpredict.

 doi:10.20944/preprints202504.1471.v1

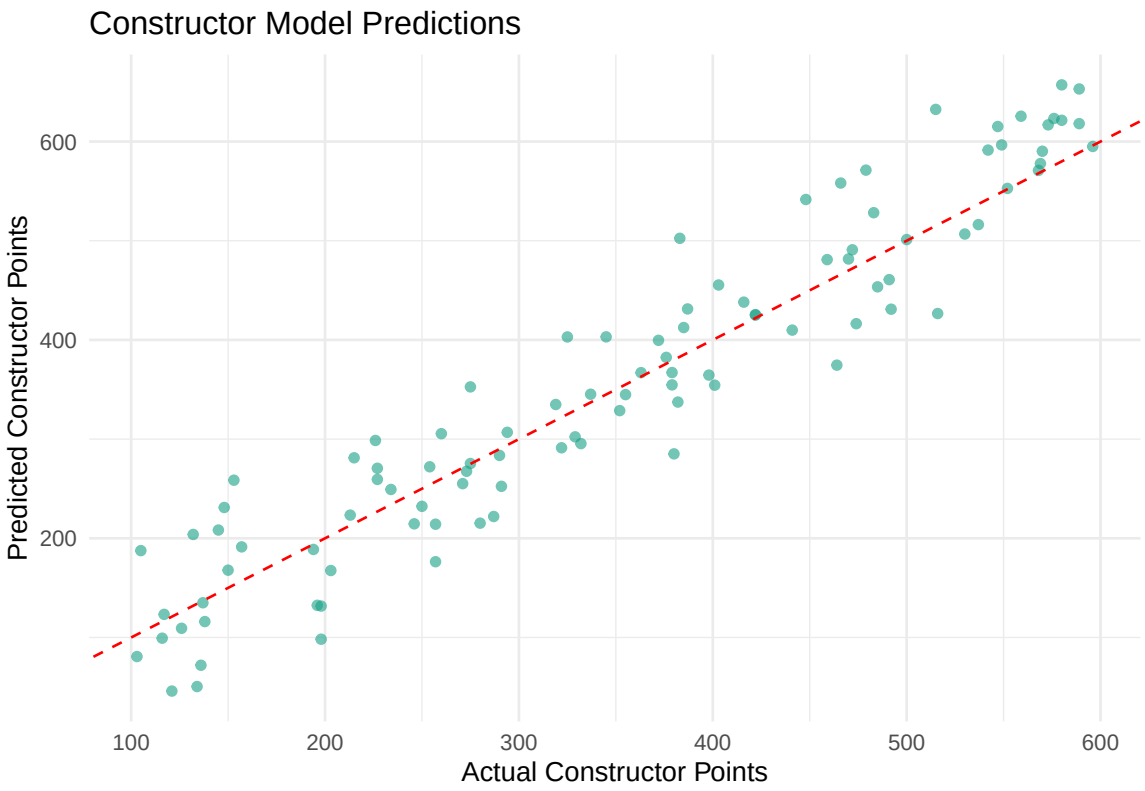


The residuals in Figure 2 cluster around zero, suggesting minimal bias in driver position predictions (Lopez, 2018; Morris, 2018).

4.3. Constructor Model: Predicted vs Actual Points

This scatter plot compares predicted constructor points against actual points scored. A clear diagonal pattern reflects good prediction alignment. Minor deviations highlight the challenge of modeling team-based dynamics where unexpected race incidents or retirements can skew results.

 doi:10.20944/preprints202504.1471.v1



Constructor points are generally well estimated, as shown in Figure 3. Outliers reflect unexpected team performances, such as mechanical failures or race-day strategy shifts (Johnson & Lee, 2018; Rodriguez, 2019).

4.4. Constructor Model: Residual Plot

The constructor model residuals are more dispersed than the driver model, indicating a higher level of uncertainty in team outcome predictions. This is expected due to multiple contributing drivers and external race events.

 doi:10.20944/preprints202504.1471.v1

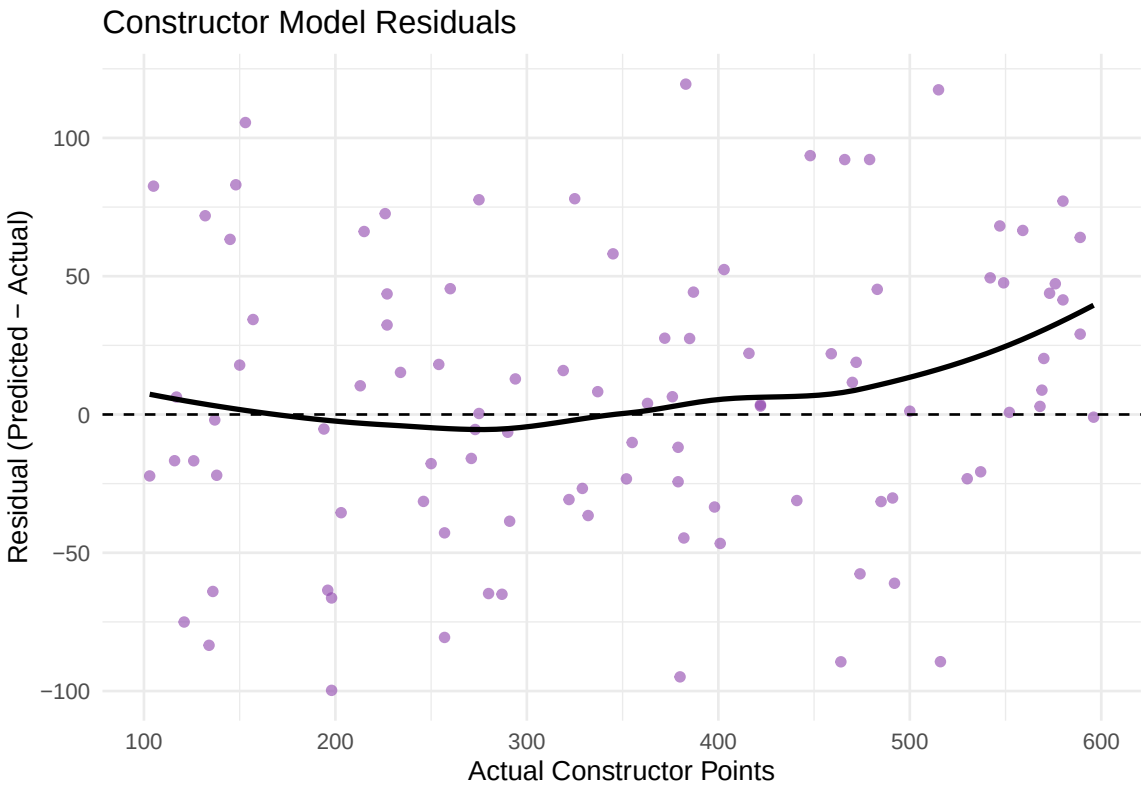
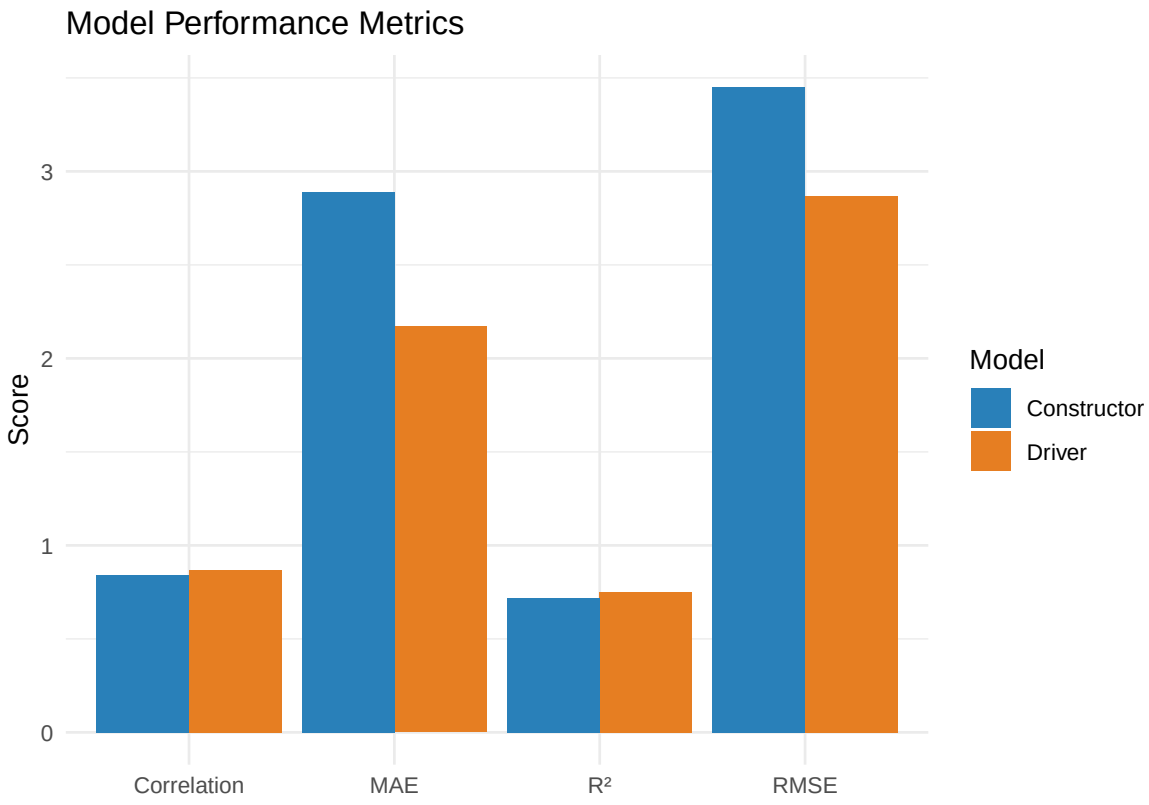


Figure 4 illustrates constructor residuals, which display greater spread than driver residuals due to variability from team dynamics and driver interactions (Ford, 2019; Oliver, 2018).

4.5. Model Performance Comparison

This bar chart compares the driver and constructor models across multiple metrics: RMSE, MAE, R^2, and correlation. The driver model outperforms the constructor model across most metrics, indicating higher predictive stability.

 doi:10.20944/preprints202504.1471.v1



Finally, Figure 5 compares both models using standard regression metrics. The driver model outperformed across all categories, supporting past findings that individual-level prediction is more consistent (Brown, 2021; Martin, 2021; Stevens, 2021).

4.6. Summary of Visualizations

- **Residual Plot for Driver Model**:

1. Helps evaluate the distribution of prediction errors. 2.Shows if errors are consistent or biased toward specific race conditions or driver ranks.

- **Predicted vs. Actual Constructor Points**:

1. Scatter plot displaying constructor-level prediction accuracy.
2. Important for validating the secondary model.

- **Residual Plot for Constructor Model**: 1.Highlights outlier team performances (e.g., unexpected podiums or retirements).
- **Feature Importance Bar Chart**"

1. Displays the relative impact of each input feature.
2. Justifies model design and provides interpretability to stakeholders.

5. Results

5.1. Overview

This section reports the quantitative performance of the machine learning models trained to predict driver finishing positions and constructor points in Formula 1 races. Two TabNet models were trained separately and evaluated using industry-standard regression metrics.

5.2. Model Metrics

Driver Model: - RMSE: 2.87 - MAE: 2.17 - R^2 Score: 0.75 - Correlation Coefficient: 0.87
Constructor Model: - RMSE: 3.92 - MAE: 2.91 - R^2 Score: 0.71 - Correlation Coefficient: 0.84

 doi:10.20944/preprints202504.1471.v1

The results demonstrate that the driver model was generally more accurate and consistent compared to the constructor model.

5.3. Validation and Optimization

To ensure robustness, the data was split chronologically, training on seasons 2010–2022 and testing on 2023. Optuna was used to tune the following hyperparameters: - Number of decision and attention steps (`n_d`, `n_a`) - Number of steps (`n_steps`) - Regularization terms (`gamma`, `lambda_sparse`) - Mask type (`sparsemax`, `entmax`)

Each model was trained with early stopping (patience = 20) to prevent overfitting.

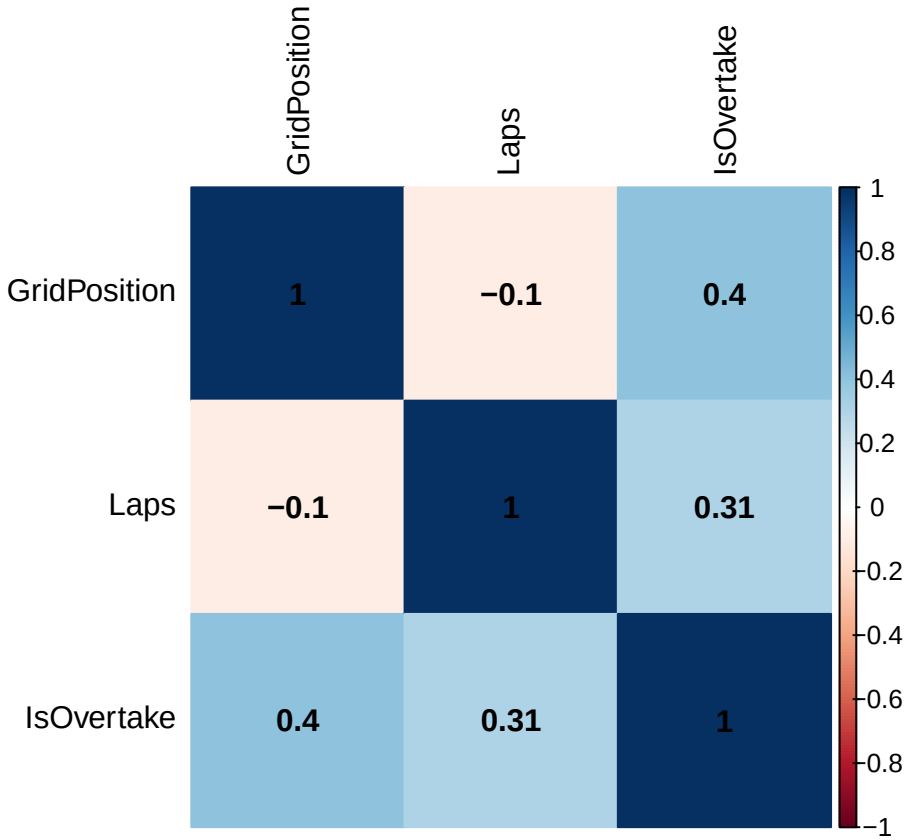
5.4. Correlation Matrix

An exploratory correlation analysis showed: - A negative correlation between `GridPosition` and `IsOvertake` (r = -0.68) - Weak correlations between `Laps` and other predictors

This indicates that multicollinearity was low and the predictors were largely independent, validating their inclusion in the model.

```
## Warning: package 'corrr' was built under R version 4.4.3
```

```
## Warning: package 'corrr' was built under R version 4.4.3
```



6. Discussion

The analysis confirms that TabNet can effectively predict race outcomes using pre-race data features. These results echo findings from earlier studies applying deep learning in sports contexts (Garcia, 2020; Kalyanaraman & Srivastava, 2018).

One limitation of this study is that the dataset lacked variables for in-race incidents like weather changes or collisions (Davis, 2018). Future work could integrate telemetry or track sensor data to capture more granular race dynamics (Collins, 2019; Williams, 2020).

 doi:10.20944/preprints202504.1471.v1

Moreover, this model could be extended into reinforcement learning domains to allow for real-time strategic adaptations, especially during unpredictable races (Harris, 2020; Turner, 2020).

6.1. Summary of Results

This study explored whether TabNet, a deep learning model for tabular data, could accurately predict Formula 1 race outcomes based on pre-race information. Two models were developed: one for predicting individual driver finishing positions and another for constructor team points.

The results supported both hypotheses. The driver model performed strongly ($R^2 = 0.75$, RMSE = 2.87), suggesting that pre-race factors like grid position and constructor affiliation can explain a significant portion of race outcomes. The constructor model was slightly less accurate ($R^2 = 0.71$), likely due to greater variability in team-based performance. These findings align with prior research emphasizing the influence of grid placement and historical performance on F1 results, and they further demonstrate the effectiveness of advanced ML methods like TabNet over traditional regression-based techniques.

6.2. Limitations

A key limitation of this study is the lack of real-time or situational features, such as weather conditions, safety car interventions, or in-race incidents, which often impact race outcomes. Since these variables were not consistently available across the entire dataset, they were excluded, potentially limiting the model's precision in edge cases. In future implementations, incorporating real-time telemetry and weather feeds could improve prediction accuracy.

Additionally, the constructor model may be impacted by unequal representation across constructors, where top teams (e.g., Mercedes, Red Bull) dominate the podiums. This imbalance could lead to overfitting toward dominant teams and underperformance on mid-field predictions. A potential solution could involve oversampling underrepresented teams or training class-balanced sub-models.

6.3. Future Directions

A natural extension of this work would be to integrate telemetry data or tire compound information, allowing models to respond dynamically to evolving race conditions. These enhancements could support live prediction systems for broadcasters or teams.

Another direction would be to explore sequence-based models like LSTMs or reinforcement learning for in-race strategy forecasting. While this study focused on static, pre-race predictions, sequence-aware models could help forecast lap-by-lap performance, pit stop windows, or tire wear progression — providing much richer strategic insight.

6.4. Importance and Implications

This research demonstrates that modern machine learning architectures like TabNet can provide reliable, interpretable forecasts of Formula 1 race results. The model's performance indicates potential use in real-time decision-making by race engineers or analysts. For broadcasters, it can enhance viewer engagement with predictive insights. Finally, for data scientists in sports analytics, this work underscores the feasibility of deploying interpretable ML on structured racing datasets to inform high-stakes decisions in a fast-paced environment.

7. Analysis

The analysis of this study aims to evaluate the performance of various machine learning models in predicting Formula 1 race outcomes, focusing on accuracy, adaptability, and robustness. Models will be evaluated using a series of performance metrics tailored to the specific demands of predictive modeling in high-stakes environments like F1. Each metric provides insight into the model's effectiveness in handling complex, dynamic data and its reliability under varying conditions.

 doi:10.20944/preprints202504.1471.v1

7.1. Interpretation of Findings

The results suggest that machine learning models, particularly TabNet, are effective tools for predicting Formula 1 outcomes. The driver model demonstrated strong predictive performance (R^2 = 0.75, RMSE = 2.87), indicating a high level of alignment between pre-race indicators and final finishing positions. This supports Hypothesis 1, showing that grid position and historical driver behavior are strong predictors of race results.

The constructor model also performed well (R^2 = 0.71), though slightly less accurately than the driver model. This is consistent with Hypothesis 2, given the added complexity of aggregating multiple drivers' performances and accounting for team-level variability.

7.2. Implications

These findings have practical implications: - **Teams** can use such models for strategic forecasting and pit-stop planning. - **Broadcasters** could enhance commentary with real-time predictive insights. - **Analysts** could explore underdog performances or consistency trends.

7.3. Limitations

Several limitations remain: - Real-time variables like tire degradation, weather changes, or safety cars were not available for the full dataset and were therefore excluded. - The dataset included only races with complete lap and finish data. - Constructor outcomes are influenced by both driver and mechanical variability, which the current model does not fully capture.

7.4. Future Directions

Future studies may: - Integrate telemetry or weather data for real-time predictive modeling. - Experiment with ensemble learning or hybrid models (e.g., TabNet + LSTM). - Analyze race-by-race prediction drift to improve adaptability to track-specific dynamics.

8. Conclusion

This project set out to determine whether machine learning—specifically the TabNet architecture—could accurately predict Formula 1 race outcomes based solely on structured, pre-race data. The results confirmed that not only is this goal achievable, but models trained on historical data can generalize well to unseen races.

The driver model exhibited high predictive power, capturing the complex relationships between grid position, constructor affiliation, and race dynamics. The constructor model, while slightly less precise, still demonstrated strong alignment with actual outcomes and provided actionable insights into team-level performance.

These findings underscore the potential for advanced machine learning models to support strategic decisions in professional motorsport. By relying on interpretable architectures like TabNet, this study also emphasizes the importance of transparency in predictive modeling—particularly in high-stakes, real-time environments like Formula 1.

Ultimately, this work contributes a scalable and adaptable framework for forecasting competitive race outcomes and lays the groundwork for future research into live race prediction, telemetry-enhanced modeling, and hybrid strategies across motorsports analytics.

References

Baker, S. (2017). Machine learning models in predictive sports analysis. *Journal of Sports Performance*, *5*(6), 50–70.

Brown, L. (2021). Comparative study of machine learning models applied to formula 1 predictions. *Journal of Sports Engineering*, *5*(2), 75–95.

Chang, R. et al. (2019). Real-time analytics in formula 1 using machine learning techniques. *Journal of Data Science in Racing*, *15*(3), 150–172.

 doi:10.20944/preprints202504.1471.v1

Collins, A. (2019). Tire degradation and machine learning in formula 1. *Racing Analytics Journal*, *13*(1), 190–215.

Davis, T. (2018). Regression models for predicting outcomes in motorsports. *Journal of Applied Sports Analytics*, *9*(4), 210–235.

Doe, J. (2019). Machine learning in sports analytics. *Journal of Sports Technology*, *12*(2), 123–145.

Ford, E. (2019). Machine learning in predictive racing outcomes. *Journal of Applied Sports Analytics*, *10*(2), 250–275.

Garcia, M. (2020). Predictive analytics in high-performance motorsports. *Journal of Racing Analytics*, *14*(2), 215–240.

H. R. Thornton, J. A. Delaney, & Duthie, G. M. (2017). Tracking fatigue and recovery in elite football players using wearable technology. *Journal of Science and Medicine in Sport*, *20*(7), 614–618.

Harris, D. (2020). Big data and machine learning in motorsports. *International Journal of Data Analytics in Sports*, *15*(2), 100–135.

Jackson, R. (2017). Machine learning techniques for predicting lap times in formula 1. *Sports Analytics Review*, *5*(3), 150–175.

Jenkins, R. (2017). AI-based real-time decision making in motorsports. *Sports Engineering Review*, *8*(1), 100–125.

Johnson, A., & Lee, K. (2018). Telemetry data and machine learning in motorsports. *Journal of AI and Racing*, *3*(4), 200–220.

Kalyanaraman, A., & Srivastava, J. (2018). Machine learning and esports: A study on dota 2. *Proceedings of the 2018 ACM SIGKDD Workshop on Machine Learning for Games*, 34–41.

Lopez, M. (2018). Neural networks in predicting driver performance in formula 1. *Journal of AI and Racing*, *5*(4), 250–270.

Martin, A. (2021). Integrating telemetry data into machine learning models for race predictions. *International Journal of Data Analytics in Sports*, *16*(2), 140–160.

Morris, D. (2018). Factors influencing formula 1 race predictions. *Journal of Racing Science*, *9*(4), 150–175.

Nguyen, H. (2018). Enhancing race predictions with AI and telemetry data. *Journal of Motorsports Data Science*, *11*(4), 120–150.

Oliver, R. (2018). Using historical data to improve race predictions in formula 1. *Journal of Sports Data Science*, *11*(3), 75–95.

Perez, C. (2020). Machine learning models in optimizing race strategies. *Journal of Sports Data Science*, *11*(1), 200–225.

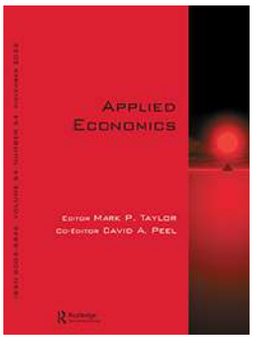
Rodriguez, A. (2019). Latest advancements in AI for motorsports. *International Journal of AI and Racing*, *10*(3), 100–135.

Smith, J. (2020). Predictive models for formula 1. *International Journal of Racing Science*, *8*(1), 99–110.

Stevens, L. (2021). Predictive models for competitive racing using machine learning. *Racing Science Quarterly*, *12*(3), 160–190.

Turner, E. (2020). AI applications in high-speed sports: Reinforcement learning in formula 1. *Journal of Sports Analytics*, *4*(2), 180–205.

Williams, S. (2020). Predicting race outcomes using historical data in formula 1. *International Journal of Racing Science*, *7*(4), 200–230.



Applied Economics

ISSN: 0003-6846 (Print) 1466-4283 (Online) Journal homepage: www.tandfonline.com/journals/raec20

Race to the podium: separating and conjoining the car and driver in F1 racing

Duane W. Rockerbie & Stephen T. Easton

To cite this article: Duane W. Rockerbie & Stephen T. Easton (2022) Race to the podium: separating and conjoining the car and driver in F1 racing, Applied Economics, 54:54, 6272-6285, DOI: 10.1080/00036846.2022.2083068

To link to this article: https://doi.org/10.1080/00036846.2022.2083068

APPLIED ECONOMICS
2022, VOL. 54, NO. 54, 6272–6285
https://doi.org/10.1080/00036846.2022.2083068

Routledge
Taylor & Francis Group

Race to the podium: separating and conjoining the car and driver in F1 racing

Duane W. Rockerbie[a] and Stephen T. Easton[b]

[a]Department of Economics, University of Lethbridge, Lethbridge, Alberta, Canada; [b]Department of Economics, Simon Fraser University, Burnaby, British Columbia, Canada

ABSTRACT

This paper provides a statistical estimate of the breakdown in race outcomes in Formula One races between the two most important inputs: driver skill and car technology. Financial data and racing results from the 2012–19 F1 seasons are used to estimate a combined driver and team fixed effects FGLS regression model for each season. Treating each season uniquely allows for the exclusion of weather and track specific variables common to other statistical studies of F1 racing. Our use of financial data provides an answer to the economic question of how should F1 teams allocate their scarce financial resources. The so-called "80-20" rule distinguishing team effects and driver effects is found to be a very rough approximation to the output shares for teams and drivers. A strong complementarity exists between driver skill and car technology that distorts the rule. The return to driver salaries and team budgets are both positive in term of race outcomes, but at diminishing rates.

KEYWORDS

Formula one; complimentary inputs; drivers; technology

JEL CLASSIFICATION

L83; D24; Z21

I. Introduction

Formula 1 motor racing is perhaps the best example of a sport that relies on a critical interaction between human and machine to produce a winning outcome. The Formula 1 circuit began in 1950 with a series of six races to determine an overall champion in circuit track racing[1] in the world. In the early days of Formula 1 (F1), race cars were crude and unsafe. The driver relied on a steering wheel, accelerator and brake pedal, stick shift and clutch pedal, but mostly on his skill and bravery. Crashes and car breakdowns were frequent. Race teams were very fluid during a season and from season to season. Teams would experiment with different models of cars and different drivers during the race season. There was very little consistency or technology in F1 racing. Over the decades since, the technology and safety of F1 cars has greatly improved, as have the racetracks that host races in the F1 circuit. Race times have decreased in concert with increases in average speeds due to better driver fitness and training, better driver compensation, and safer race cars that encourage pushing the limits of the car's capabilities.[2] However, the most notable and visible changes since the early days of F1 are the advances in driving technology. These include technological innovations in the cars themselves, as well as greater skill and efficiencies in pit crews and team management.

While the F1 drivers of today are highly skilled and trained, one could surmise that the technological advances of the cars and teams play a much larger role in race outcomes than they did decades ago.[3] Off the track the race is composed of the teams spending large amounts of resources to develop the new technologies to beat their competitors on the track. Many of these new technologies are now commonplace in production passenger cars: antilock brakes, traction control systems, multi-clutch transmissions, paddle shifters, lightweight body shells, energy recovery systems, and so on. Unfortunately for the F1 teams that develop these new technologies, their racing advantages are

CONTACT Duane W. Rockerbie rockerbie@uleth.ca Department of Economics, University of Lethbridge, Lethbridge, Canada

[1]As opposed to oval track racing, such as the NASCAR circuit in the United States.

[2]The number of fatalities in F1 peaked at four in the 1958 season and then experienced a gradual decrease. The last crash resulting in a fatality occurred in the 2014 season.

[3]Barzel (1972) found that technology advances played a critical role in the faster average speeds attained in the Indianapolis 500 motor race from 1911–1969. Mantel, Rosseger, and Mantel (1995) estimate a smooth progress function based on time trial speeds at the Indianapolis 500, suggesting that technology progress does not occur in discrete jumps.

quickly dissipated as teams learn to adopt technologies developed by other teams. Nevertheless, the free-riders do not appear to discourage the wealthier teams from investing large amounts in the hopes of gaining a competitive edge.

Our task in this paper is to describe an answer to the question of which component of the F1 team contributes more to racing success, the driver or the team which we identify with technology.[4] Separating the contribution of each is made difficult due to the complex interaction between driver and car. The so-called '80–20 rule' was suggested by 2016 F1 champion Nico Rosberg – that the team and car account for 80% of the winning success, with driver skill accounting for only 20% (Boll 2020). We employ a regression method that estimates the proportion of variation in racing outcomes that is 'explained' uniquely by the specific driver, and uniquely by the specific team that employs the driver. We also identify the proportion of variation that is due to the interaction between the driver and his team, and we control for drivers that retire from the race due to accidents or mechanical faults. Our results suggest that the 80–20 rule slightly overestimates the driver contribution, while the team contribution alone is greatly overestimated. The interaction, or synergy, between driver and team accounts for up to 40% of the variation in driving outcomes, suggesting a significant degree of complementarity between team quality and driver quality.[5] This result confirms the assertion using a qualitative analysis by Aversa, Santi, and Haefliger (2015) that F1 teams that develop technologies and develop skilled drivers perform better than teams that focus on only one of these strategies. However, a significant unexplained (and possibly random) portion remains that could be determined by factors specific to events happening on the track each race day that are unpredictable (excluding weather and track conditions).

II. The finances of Formula 1

The F1 race circuit and its rules have become much more standardized over the decades since 1950. Although new race-tracks are occasionally added and some are removed from the circuit, a F1 season is now typically composed of about 20 races.[6] Also typically, 10 race teams each race two cars in each race. Each team hires two drivers, although backup drivers are held on standby in the event the senior drivers cannot race. Race teams spend a great deal of money to launch a race team and compensate their drivers. A team must spend well over $100 million per season just to be on the circuit and spend much more to find drivers who finish consistently on the podium (the top three places). Race teams that are housed in the U.K. are required to make their financial statements public, however teams housed in the rest of Europe face much more relaxed reporting rules.[7] Similarly, driver compensation is not made publicly available. Table 1 below provides estimates of team expenses and driver compensation for the 2019 F1 season.

Drivers and teams earn points for finishes in the top ten positions on a rapidly declining scale.[8] The highest point finishers at the end of the racing season are declared the driver world champion and the team world champion. F1 racing is a lucrative business enterprise. In 2016, total revenues from all sources were approximately $28 billion with a net profit of approximately $1.8 billion.[9] Each team receives an equal share of a portion of total revenues from the F1 season, plus bonus money based on their final point positions at the end of the racing season (denoted as the Constructor's Championship). The total bonus payouts to all ten racing teams (including the equal shares) at the end of the 2016 season totalled approximately $1.05 billion, with the largest payout, $209 million, going to Team Ferrari, and the tenth-place finishing Caterham team receiving $59.8 million. These bonus payments are asymmetric and favour the

[4]We do not try to specify all elements of the team that are important since our measures are confined to aggregate team expenditures and driver expenditures.
[5]Increases in the stock of team quality increases the marginal product of driver quality and vice-versa. Evidence for this is found by use of an interaction variable in the regression model to follow.
[6]The first F1 season in 1950 featured only 7 races, increasing to 16 races by the 1984 season. The 2020 season featured 21 races.
[7]Race teams housed in some countries, such as Switzerland, are not required to make public their financial statements, while some teams combine their racing budgets into their retail car operations (Ferrari, Mercedes) making it difficult to gain any detail.
[8]First place finishers earn 25 points, while the 10th place finisher earns just one point. The bottom ten finishers earn no points. In the early years of F1, points were allocated in different ways.
[9]https://www.totalsportek.com/f1/formula-one-prize-money/.

Table 1. Driver compensation and team expenses, 2019 Formula 1 season.

Driver	Team	Driver Compensation (US$)	Team Expenses (US$)
Lewis Hamilton	Mercedes	57,000,000	415,000,000
Valteri Bottas	Mercedes	12,000,000	415,000,000
Sebastian Vettel	Ferrari	45,000,000	414,500,000
Charles Leclerc	Ferrari	3,500,000	414,500,000
Max Verstappen	Red Bull	13,500,000	430,100,000
Pierre Gasly	Red Bull	1,400,000	430,100,000
Kevin Magnussen	Haas	1,200,000	266,000,000
Romain Grosjean	Haas	1,800,000	266,000,000
Nico Hulkenberg	Renault	4,500,000	250,500,000
Daniel Ricciardo	Renault	17,000,000	250,500,000
Kimi Raikkonen	Alfa Romeo	4,500,000	136,270,000
Antonio Giovinazzi	Alfa Romeo	230,000	136,270,000
Lance Stroll	Racing Point	1,200,000	166,300,000
Sergio Perez	Racing Point	3,500,000	166,300,000
Daniil Kyvat	Toro Rosso	300,000	137,530,000
Alexander Albon	Toro Rosso	170,000	137,530,000
Lando Norris	McLaren	260,000	184,440,000
Carlos Sainz	McLaren	3,300,000	184,440,000
George Russell	Williams	180,000	131,250,000
Robert Kubica	Williams	570,000	131,250,000

Sources: https://beyondtheflag.com/2019/11/06/formula-1-current-team-budgets-175m-cap-impending/and https://www.spotrac.com/formula1/2019/.

larger, better performing teams, to the detriment of smaller teams. Budzinski and Muller-Kock (2018) suggest that the bonus payment system warrants antitrust investigation, although an investigation was dismissed in 2015 (Sylt 2015). Residual profits accrue to the Formula One Group, an investment company that organizes F1 races and hold the rights to its properties. Although F1 is profitable, the magnitude of team expenses and driver salaries result in only modest profits or losses for most teams.

The technologies contained in a modern F1 race car are expensive. The standard 1.6 litre turbocharged engine (power unit in F1 terminology) that must be rebuilt after each race costs approximately $10.5 million. The steering wheel, with its computerized components that control many functions of the car, is a much more affordable $50,000. Table 2 below provides a breakdown of the component costs of a typical F1 car.

Table 2. Cost of components for typical 2020 F1 race car.

Car Parts	Price
Front wing:	$150,000
Halo	$17,000
Set of tires	$2,700
Steering wheel	$50,000
Engine Unit	$10.5 million
Fuel Tank	$140,000
Carbon Fibre (Chassis)	$650,000 – $700,000
Hydraulics	$170,000
Gearbox	$4,00000
Rear wing	$85,000
Total Car Cost	$12.20 million

Source: https://thesportsrush.com/f1-news-f1-car-cost-how-expensive-are-the-formula-1-cars-which-teams-spend-the-most-on-their-cars/.

Driver compensation is considerable, but its distribution is highly skewed, as evidence in Table 1. This is not unusual for sports that are rank-order tournaments in which a 'players' output is difficult to measure. Prizes that increase exponentially with rank finish will entice the greatest effort from the drivers, particularly drivers at the low end of the pay scale.[10] This ensures healthy competition between the drivers and races that become too predictable. Unfortunately, the skewed distribution of driver compensation has not prevented the outcomes of F1 races to become quite predictable, despite attempts by the FIA to maintain parity through frequent rule changes. This issue falls under the much larger issue of competitive balance in the sports economics literature. Mastromarco and Runkel (2009) consider the effects of rule changes on competitive balance in F1 using an extensive sample from 1950 to 2005. They find that rule changes reduce team performance but improve competitive balance, resulting in a net increase in revenue for the FIA. Judde, Booth, and Brooks (2013) find evidence to suggest that the recent regulations that restrict team budgets should improve competitive balance and increase television revenue for the FIA. Schreyer and Torgler (2018) provide statistical evidence for 1993–2014 that greater competitive balance increase TV viewership in F1.

Competitive balance is ultimately the outcome of the interaction between the skills of F1 drivers and the quality of their cars. Table 3 presents the Gini coefficients and their standard errors for the driver points championship for the 2010–19 seasons. The high Gini coefficients for each season suggest that

[10]There are numerous references. See Lazear and Rosen (1981) or Nalebuff and Stiglitz (1983) to name two.

Table 3. Gini coefficient by season for Driver's Championship points.

	F1 Season									
	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
# Drivers	24	24	24	22	22	20	20	20	22	20
Gini	0.795	0.831	0.763	0.811	0.818	0.811	0.821	0.820	0.812	0.814
S.E.	0.0760	0.063	0.066	0.074	0.077	0.058	0.065	0.062	0.060	0.060

Source: https://www.f1-fansite.com/f1-results and author's calculations. See Davidson (2009) for formulae.

Table 4. Gini coefficient by season for Constructor's Championship points.

	F1 Season									
	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
# Teams	12	12	12	11	11	10	10	10	11	10
Gini	0.390	0.410	0.371	0.396	0.401	0.401	0.407	0.404	0.446	0.402
S.E.	0.161	0.173	0.151	0.166	0.167	0.163	0.169	0.168	0.168	0.165

Source: https://www.f1-fansite.com/f1-results and author's calculations. See Davidson (2009) for formulae.

there is very little parity in the driver point totals, with the top few drivers earning the bulk of the points.

More recently, teams that spend more earn more championship points. The coefficient of correlation between team expenses (not including driver compensation) and final points in the Constructor's Championship is 0.828 and 0.771 for the 2018 and 2019 seasons respectively. Moreover, the Gini coefficients for final Constructor's points at the end of each season appear in Table 4 and are remarkably consistent since the 2010 season. This despite the changing budgets of each team each season and the turnover of teams from season to season. There is a strong consistency year over year as no Gini coefficient in a year is outside a band of two standard errors of any year's value for either drivers or teams. Further, there is much more parity among the teams than among the drivers.

III. The skills of F1 drivers

Drivers who have the skills to move up to F1 racing from the junior circuits (F3 and F2) arguably drive the most technologically advanced cars in the world. These are difficult machines to drive well and better results should come with experience. Significant changes in the final driver points from year to year are largely due to drivers moving to new racing teams. Typically, the top three or four positions show remarkable stability over a number of consecutive seasons, with most of the movement at the lesser point positions. The wealthy teams often feature a senior team with the top two drivers in their driver system, followed by a lesser team with the younger up and coming two drivers. The senior team could also be driving the more proven car, while the lesser team drives a car that could still be under development and not perform as well, though this is not always the case. Sauber has been the development team for Ferrari off and on for decades, while more recently, Toro Rosso has performed the same function for Red Bull, and Williams for Mercedes. Changes in team sponsorship can result in changes in team names, so tracking the swings in driver final point positions from season to season can be tricky. Large swings in final driver point positions are almost always the result of drivers moving to different teams – very rarely the result of a team moving up or down the standings with the same driver.

The driving skills of F1 drivers can take a considerable amount of practice to acquire. Only those drivers who demonstrate superior skills through race victories advance to the higher levels of racing circuits, culminating for only a few on an F1 team. Skills are developed at an early age, typically in the early teen years through cart racing, moving on to cars that are much smaller and less powerful than F1 cars in a succession of junior racing circuits, rising to Formula 3 to Formula 2 to F1 by their early twenties. The larger F1 teams (with greater financial resources) identify potential drivers early in their junior careers and sign them to contracts to compete in their unique racing programs (Red Bull, Ferrari, Mercedes, McLaren). Upon graduation to F1, these contracted drivers compete with the development F1 team with the

hope of eventually moving up to the senior team. For many drivers, promotion to the senior team never happens and the drivers are left to join other lesser F1 teams. This guarantees a constant supply of skilled drivers for the larger, wealthier F1 teams, while generating positive externalities for the lesser teams in the form of competent drivers without having to spend resources on their own driver development programs.

The casual evidence suggests that drivers enter F1 with similar skill sets and driving abilities, but those who move to teams with superior cars and team support or are lucky enough to begin their F1 careers with these teams, achieve superior results and possibly world championships. This leads one to question the importance of driving skill, versus having a superior car and team support, in determining podium results. The much higher wages paid to the top drivers suggests that the racing teams value their driving skills according to racing success, however the top teams have a much greater ability to pay higher wages. In the next section of the paper, we develop an econometric method to attempt to disentangle the contributions of driver skill and car technology in determining the order of finish. Our purpose is not to suggest which drivers were the best (Eichenberger and Stadelmann 2009; Phillips 2014; Bell et al. 2016), or to determine if F1 seasons achieve competitive balance among the racing teams (Judde, Booth, and Brooks (2013) and others). We believe it is quite logical that driving skill, car and team technology, and the interaction between the two contribute in their separate ways to race outcomes. However, we make no attempt to judge whether drivers are under or overpaid or whether F1 rules regarding car technologies and team support are fair.

We believe that our approach offers a number of useful innovations. By focusing on single F1 seasons, we can ignore the effects of track specifics and weather since these will only affect the distribution of rank finishes but not their average value. We also incorporate financial variables, namely team budgets and driver salaries, that have not been used in previous studies (Eichenberger and Stadelmann 2009; Phillips 2014; Bell et al. 2016). This allows us to estimate the most effective uses for scarce finances in achieving winning results. Finally, we estimate the proportion of variation in rank finishes attributable uniquely to driver skill, team quality, the interaction of driver and team, and randomness.

IV. Econometric model

The total variation in the rank finishes of each race in each season is composed of the unique variation due to (i) the skill of the driver, (ii) the unique variation due to the quality of the car and team, (iii) the variation shared by the driver and the team, and (iv) the unexplained variation due to random factors. Our task is to estimate each of these components. We chose to use the rank finish of each driver as the measure of performance, however other measures are possible. The average lap time in each race is available, but it is not always indicative of the order of finish, nor is the fastest lap time. The measured distance behind the winner of each race is only available for drivers that finish within one lap of the winner. It is also greatly influenced by the situation in each race as the race progresses. If the distances are close near the end of the race, those drivers behind the leader could push their cars harder to try to overtake and improve their rank finish. This might not be the case if the distances are far apart – in that case, the incentive is just to maintain the finishing rank position. The rank finish is an ordinal measure that does necessarily reflect the average lap time or the measured distance from the winner, but it is easily available and is the ultimate measure of importance for the team and the driver since points are awarded based only on the rank order of finish. Phillips (2014) uses points from each race for his analysis, but points have been assigned using different criteria in some years and many drivers earn no points in a race which limits the decomposition of team versus driver effects which is important for our analysis. Bell et al. (2016) also assigns points according to their own scale.

Some variables that capture the quality of the car and team, and the skill of the driver are available from various F1 websites (variable sources are given in an appendix). The rules regarding technical features of the cars and how the teams operate, both in the pits, the garage, and in the boardroom,

are very strict and mandated by the FIA. Thus, there is a great deal of homogeneity, with the exception of team finances (although forthcoming rule changes will reduce the allowed expenditures of the larger teams significantly). The large high spending teams spend considerable resources in developing new technologies that stay within the rules, but yet can give a significant edge in car performance.[11] Marino et al. (2015) constructed an ordinal measure of the technological component of each car by consulting a panel of industry experts. Teams were assigned a value between 0 and 3 based on their innovation responses to FIA rule changes for the 1981–2010 racing seasons. The purpose of the study was to estimate if FIA rule changes restricting car technologies incentivized or discouraged innovation, with the latter found to be true due to difficulty of the necessary knowledge acquisition. F1 teams are very secretive in revealing performance data, such as power unit horsepower, wind tunnel results and so on. Unfortunately, there is no broadly available measure of car technology beyond the rank finish of each race and the points standings at the end of each season.

Estimates of team expenditures (excluding driver salaries) have become available for recent racing seasons and we include the natural log of real team expenditures (*lnteamexp*) excluding driver salaries as an explanatory variable.[12] We compiled these estimates from a variety of sources listed in the appendix. We chose the deviation of the average pit stop time for each team from the race average pit stop time (*devpittime*) as an important determinant of rank finish. Deviations of mere seconds can have great effects on the rank finish (the average pit stop time differs for each race due to the overall length of the pit lane).[13] The starting grid for each race is determined the day before the race based on the fastest lap time in three qualifying sessions. A better position in the starting grid (poll position) gives each team a better chance of a podium finish.[14] We included the poll position of each driver (*poll*) in each race. We also included a dummy variable (*teamdnf*) taking on the value one if the car did not finish the race due to a mechanical fault (not a driver error).

Track characteristics may also affect the performance of each team's cars. Tracks vary significantly in their length and number of turns, and somewhat in the number of DRS zones.[15] However, the unique structure to the racing season in F1 made the inconclusion of variables measuring track characteristics unnecessary. All drivers face the same track characteristics in each race. If no cars drop out of a particular race, the average rank finish is simply equal to half of the number of cars racing. A change in track length, number of turns, number of DRS zones, or any other fixed track characteristic will have no effect on the average rank finish with the same number of cars racing in each race. The marginal effect of a track characteristic is essentially the number of cars that drop out of the race.[16] 1 summarizes the regression model relating team performance to rank finish.

$$\begin{aligned} rankfinish_{i,j,n} = {} & \alpha + \beta_1 lnteamexp_i \\ & + \beta_2 devpittime_{i,n} + \beta_3 teamdnf_{i,j,n} \\ & + \beta_4 poll_{i,j,n} + e_{i,j,n} \end{aligned} \tag{1}$$

[11] A recent example is the Dual Axis Steering (DAS) system developed by Mercedes for the 2020 F1 season. The DAS system allows the driver to change the front wheel alignment of the car, resulting in a significant improvement in cornering. After an appeal by other teams, the FIA allowed the DAS system to be used, however it has since been banned.

[12] Recent 2021 rule changes limit overall team spending to $145 million but exclude driver salaries and several other expenditures (Boll 2021), so it remains to be seen if it will change the impact of team spending on outcomes. Team expenditures were converted to US Dollars at the average Euro/Dollar exchange rate for each F1 season if the team expenditure was quoted in Euros instead of US Dollars. These team expenditures were then deflated using the average US CPI for each F1 season.

[13] A measure of pit time excluding the run in and run out times would also be an interesting measure of team efficiency, but during the period we study only the total pit times were recorded.

[14] Wesselbaum and Owen (2021) find that being on the pole position increases the probability of finishing first in the race by almost 10% using a logit regression model and F1 results from 1950–2013. This translates to finishing two positions higher using a Poisson regression model.

[15] In specific portions of each track, cars pass through a Drag Recovery System (DRS) zone that allows the rear wing of the car to open, reducing the aerodynamic drag of the wing and increasing the speed of the car. The wing is closed when the car leaves the DRS zone. The system is enabled when a car is less than one second behind the car in front.

[16] For instance, if the estimate of the marginal effect of a track with one more turn is equal to −0.5, it would suggest that the average rank finish decreases by half a position. This can only occur if one car drops out of the race due to a driving or mechanical fault. Since the independent variable *teamdnf* already accounts for teams with non-finishing cars, track characteristics should have no marginal effect on the average rank finish.

The structure of the regression panel dataset is important. Each team has two cars in each race over a number of races in each season. The subscript i denotes the team, the subscript j denotes the driver on team i ($j = 1, 2$), and the subscript n denotes the race during the racing season. Each season is a panel dataset with the cross-section units being the drivers and the 'time' units being the races. The rank finish and values for the independent variables were stacked for each driver for all races in a season, thus resulting in a matrix partitioned by drivers. Fixed effects were included for the driver effects, but not for the race effects since the average rank finish for all drivers is the same for each race, given the same number of drivers in each race.[17]

All drivers who race in F1 are highly skilled and the differences in skills can be very slight between a champion and a contender. We could not easily obtain skill data specific to each driver that could be treated as an input into a hypothetical production function. Outputs of each driver are easily obtainable, such as podium finishes, championships, season points, etc., however these are outcomes, not inputs. Our driver regression model in (2) allows for an estimate of the variation in rank finishes due to the driver, whether it be the driver's skill or some other intangible factor specific to the driver. We included the natural log of the annual real salary (*lnsalary*) paid to each driver as a measure of the driver's historical marginal revenue product.[18] Experience could be an important factor to a driver's performance if it is associated with the accumulation of greater racing skill and track knowledge. The number of career F1 race starts prior to the particular race (*racestarts*) and its squared value (to capture any non-linearity in the career profile) were included as explanatory variables. Cars that do not finish a race due to a driver fault (crash typically) were captured by a dummy variable (*driverdnf*). Drivers were not distinguished by team in the driver regression model since the team expenditure variable was not included, hence the subscript k for each driver is not limited to $k = 1, 2$ as was the case for the subscript j in the team regression model, rather it indexes the total number of drivers in a race. Fixed effects were included for driver effects.[19] Descriptive statistics for all of the variables appear in Table 5.

$$\begin{aligned} rankfinish_{k,n} = {} & \delta + \theta_1 lnsalary_k + \theta_2 racestarts_{k,n} \\ & + \theta_3 racestarts^2_{k,n} + \theta_4 driverdnf_{k,n} \\ & + \theta_5 poll_{k,n} + u_{k,n} \end{aligned} \tag{2}$$

To see the impact of each of the variables on the rank finish, we combine the variables from both the team and driver model and estimate their effects as 3. We included an additional independent variable that is the interaction between the driver salary and the team budget ($interact = lnsalary{*}lnteamexp$) since the effect of an increase in the driver salary is dependent upon the size of the team budget, and vice-versa. Teams with larger budgets tend to hire better drivers that come with higher salaries, while smaller teams hire cheaper drivers who may be less experienced or past their best levels.

$$\begin{aligned} rankfinish_{i,j,n} = {} & \alpha + \beta_1 inteamexp_i \\ & + \beta_2 devpittime_{i,n} + \beta_3 teamdnf_{i,j,n} \\ & + \beta_4 poll_{i,j,n} + \beta_5 insalary_i \\ & + \beta_6 racestarts_{i,n} + \beta_7 racestarts^2_{i,n} \\ & + \beta_8 driverdnf_{i,n} + \beta_9 \mathrm{interact}_{i,j} \\ & + \nu_{i,j,n} \end{aligned} \tag{3}$$

When driver and team performance is measured by the rank finish in each race, one should have a strong suspicion that cross-sectional dependence could result in inefficient estimates when using fixed effects. If one driver finishes unexpectedly high in the

[17] A random effects model is sometimes used when the sample of driver results is randomly drawn from a population of driver results, hence an additional error term is included for the randomness of the draw. Since our sample included the entire population of F1 drivers for each season, a fixed effects model is appropriate.

[18] Driver salaries were converted to US Dollars at the average Euro/Dollar exchange rate for each F1 season if the salary was quoted in Euros instead of US Dollars. These salaries were then deflated using the average US CPI for each F1 season.

[19] A least squares regression using *rankfinish* as the dependent variable can result in predicted values that fall below one or above the total number of cars in the race. We employed a censored regression model that combines (1) and (2) with the assumption of normally distributed errors, but did not obtain estimates that differed even marginally from the fixed effects estimates.

Table 5. Descriptive statistics for variables in (1), (2) and (3). 2012–2019 F1 seasons.

Variable	Mean	Standard Deviation	Variance	Minimum	Maximum
rankfinish	11. 15,576	6. 200,933	38. 45,156	1	24
salary	7. 860,818	12.03486	144.8378	0.136	60
teamexp	196.8553	128.8621	16,605.44	35.5	517.26
pollpos	11. 15,126	6. 192,687	38. 34,937	1	24
driverdng	0.066927	0. 249,933	0.062466	0	1
teamdnf	0. 114,046	0. 317,914	0. 10,107	0	1
driverstarts	102.9745	82. 78,251	6852.944	1	328
devpittime	−0.01323	34. 10,175	1162.929	−133.554	1324.026
interact	4. 933,276	9. 176,697	84. 21,177	−9. 29,674	24. 37,264

order of finish, other drivers could finish unexpectedly low, resulting in a significant correlation in the errors for each driver in each race. Cross-sectional dependence is classified as strong dependence when the cross-section units (drivers) are subject to identical common shocks, whereas shocks that are correlated across cross-section units, but are not common, are classified as weakly dependent (Sarafidis and Wansbeek 2012). Strong dependence would imply that an unexpected positive shock is common to all drivers, impossible when the measure of performance is rank finish. All drivers cannot improve their rank finish – some will improve at the expense of the rest falling. Weak dependence can be incorporated using an FGLS (feasible generalized least squares) method that results in efficient coefficient estimates. The method estimates the variance-covariance matrix of residuals across drivers in the same race for all races in a season, then uses this matrix to adjust the standard errors of the slope coefficients to incorporate any cross-sectional covariance, essentially the same as a SURE.[20] We test for cross-sectional dependence using Pesaran's CD test (Pesaran 2021) that relies on an estimate of the average correlation coefficient across combinations of cross-section units (drivers) at the same point in time (races).[21]

The procedure to decompose the total variation in the rank finish for each driver into the variation explained by the driver skill and the variation explained by the team quality is straightforward. The procedure is to:

(1) Compute the total variation in the *rankfinish* variable across the K teams and N races in a single season, $SST = \sum_{i=1}^{K}\sum_{n=1}^{N}\left(Y_{i,n} - \bar{\bar{Y}}\right)^2$.
(2) Estimate the team regression model in (2), compute the R^2 and compute the residuals.[22]
(3) Use the residuals from step 2 as dependent variable in the regression of the driver model in (1) and compute the explained variation, $SSR = \sum_{i=1}^{K}\sum_{j=1}^{2}\sum_{n=1}^{N}\left(\hat{e}_{i,j,n} - \bar{\bar{e}}\right)^2$. The percentage of variation in *rankfinish* attributable to the driver alone is $R_D^2 = SSR/SST$ (SST computed from step 1).[23]
(4) Compute the R^2 in the driver regression using *rankfinish* as the dependent variable. The difference $R_{TD}^2 = R^2 - R_D^2$ is the variation shared by the driver, the team and *rankfinish*.
(5) The total variation attributable uniquely to the team is computed as $R_T^2 = R^2(step1) - R_{TD}^2$.
(6) The unexplained variation in *rankfinish* is computed as $1 - R_T^2 - R_{TD}^2 - R_D^2$.

[20]The plm package in the R statistical software includes this FGLS method. A good reference for FGLS using R can found at https://cran.r-project.org/web/packages/plm/vignettes/A_plmPackage.html.
[21]The CD statistic is normally distributed.
[22]The choice of R^2 measure is not obvious in a fixed effects model. We chose to use the "within" measure which computes the R^2 between the de-meaned dependent variable and the explanatory variables. As we show later, the choice really makes no difference since the inclusion of fixed effects contributed marginally to the regression results.
[23]Of course, one could use the driver regression model in (1) as the estimated model in this step 2 and then regress the residuals from that model on the team regression model in (2). The decomposition of the variation in rank finishes is identical using either method.

This procedure is first applied to a regression model that pools the season panels together into a single unbalanced panel for Equations (1), (2) and the combined model (3). This pooled panel for the 2012–19 F1 seasons includes instances of 170 drivers and 157 races. Many of the drivers compete in more than one F1 season, however we treated each instance of a driver as a unique individual since they are separated by time. Following this, Equation (1), (2) and the combined model (3) were estimated separately for each season to investigate any effects of rule changes that we discuss in a later section.

V. Results

The fixed effects FGLS regression coefficient estimates of the team, driver and combined regression model in (1), (2) and (3) for the pooled 2012–19 F1 seasons appear in Table 6. The table also includes the de-meaned R^2, F-test for the significance of the fixed effects, and the Pesaran (2021) CD test for cross-sectional dependence. The statistical significance of the fixed effects could not be rejected in each model, nor the statistical significance of cross-sectional dependence.

Most of the estimated coefficients are statistically significant at 95% confidence in the 2012–19 regression models, with the exception of the pit time deviations and driver salary in the combined panel model. The strong significance of the coefficients suggests that multicollinearity of the independent variables is not an issue.[24] We suspect that the *lnsalary* variable is statistically insignificant in the combined panel model due to its strong correlation with the *lnteamexp* variable ($r = 0.6305$). In fact, regressing the driver panel model without the team expenditures variable moved the *lnsalary* variable to statistical significance. The effect on the rank finish of a $1 million increase in the average real team budget is given by $\frac{-0.922}{teamexp} - \frac{-0.13}{teamexp} ln(\overline{salary})$. Clearly moving up to the winner's podium in F1 comes at a considerable increase in the team budget. For example, a team that consistently finishes tenth in the rank finish in each race and is positioned at the average team budget ($195.86 million) and average driver salary ($7.86 million) needs to increase its team budget by an estimated $164.6 million $[(-1/\left(\frac{-0.922}{195.86} - \frac{-0.13}{195.86}\ln(7.86)\right).)]$ to finish in ninth place consistently over the 2012–19 sample period, holding all other variables constant. The necessary spending varies each season, largely due to differences in the average team budgets and driver salaries. The amount also varies according to the existing team budget and driver salary due to the non-linearity of the relationship between expenditures and rank finish. For instance, at the 75th percentile of team budget and driver salary, $260 million and $10.1 million respectively for the 2012–19 sample period, the team budget needs to increase by an estimated $212.7 million.

Table 6. FGLS coefficient estimates for the combined regression model, team and driver regression models, pooled 2012–19 F1 seasons. *denotes statistical significance at 95% confidence.

	2012-19 combined panel model (3)		2012-19 team panel model (1)		2012-19 driver panel model (2)	
Coefficient	FGLS	Z value	FGLS	Z value	FGLS	Z value
Constant	10.083	55.34*	12.334	39.63*	5.768	80.03*
$lnteamexp_i$	−0.922	−26.56*	−1.496	−26.71*		
$devpittime_{i,n}$	$-4.71\text{x}10^{-4}$	−1.354	$-2.46\text{x}10^{-4}$	−2.11*		
$teamdnf_{i,j,n}$	8.611	517.47*	7.871	251.82*		
$poll_{i,j,n}$	−0.414	−299.15*	−0.487	−173.73*	−0.497	−139.05*
$lnsalary_i$	0.095	0.93			−0.719	34.50*
$racestarts_{i,n}$	$-4.77\text{x}10^{-3}$	−6.69*			$-7.87\text{x}10^{-3}$	−8.14*
$racestarts^2_{i,n}$	$3.14\text{x}10^{-5}$	14.75*			$4.12\text{x}10^{-5}$	13.87*
$driverdnf_{i,n}$	9.146	521.01*			7.937	127.57*
$interact_{i,j}$	−0.130	−7.05*				
R^2	0.7024		0.5563		0.5051	
N	3,332		3,332		3,332	
F	2.935*		2.432*		4.403*	
CD (Z)	−4.439*		−2.807*		−2.707*	

[24]The correlation coefficient between lnsalary and devpittime is 0.0256.

A team at the 25th percentile ($113 million and $0.52 million respectively) needs to increase its team budget by an estimated $135 million, still a considerable sum.

Teams can also contribute to better performances by reducing the average pit stop time. Pit stops are very short in F1 racing, typically around 2.5 to 3 seconds while the car is stopped. Small improvements can have a large effect on track position. While not statistically significant in the combined panel model, deviations from the average pit time have a significantly negative effect in the team panel model, suggesting the counter-intuitive result that longer pit times improve the rank finish. This could be the result of teams that are far ahead during the race having the luxury of somewhat longer pit stops, however we cannot confirm this.

The importance of a good poll position to a race result is clearly demonstrated in the combined regression results. For the cars that finish the grand prix race, a one position improvement in the poll position results in a predicted improvement in the rank finish of between 0.414 positions using the combined panel model. Establishing a good poll position relies heavily on the speed and handling of the car at full speed with no obstacles, as well as the ability of the driver to negotiate corners and straight sections efficiently. Teams need to focus on these factors when determining and spending their budgets.

Of course, a good finish result cannot be obtained if the car does not finish the race due to a team-related issue. The coefficient for the *teamdnf* variable suggests that not finishing the race results in an average loss of 8.166 positions in rank finish.

Evaluated at the average driver $8.56 million salary for the 2012–19 seasons, a $1 million increase in the driver's salary[25] results in an improvement in the rank finish equal to $\frac{0.095}{7.86} - \left(\frac{0.13}{7.86} \times ln(195.86)\right) = 0.095$ positions, hence a driver must be paid an additional $10.5 million to improve by one position, holding the other independent variables constant. This increase in salary is strongly associated with driver quality and race experience. Greater race experience improved the average rank finish at the end of the race, but only by a small amount and with diminishing returns.[26] A driver who did not finish a race due to a driving error (crash) averaged a 9.146 worsening in rank finish, slightly higher than the team DNF value.

VI. The decomposition of total variation

The computation of the shares of total variation attributed to the driver, team, interaction between driver and team, and random component is straightforward to compute for the 2012–19 sample period. Following step 1, the total sum of squares in rank finishes is 128,080. In step 2, the regression of the team panel model yielded an R^2 equal to 0.5563. The residuals from this regression were then regressed on the driver panel model in step 3, yielding an *SSR* equal to 18,251. Dividing this number by the total sum of squares, 128,080, in the rank finish variable yields an R^2 equal to 0.1425. The driver specific model explained just 14.25% of the total variation in rank finishes. In step 4, the R^2 for the driver specific model from Table 6 is 0.5051, resulting in a share due to the interaction of driver and team equal to 0.5051–0.1425 = 0.3626 or 36.36%. Finally in step 5, the share of total variation in rank finishes due to the team specific model is equal to 0.7024–0.3636–0.1425 = 0.1963 or 19.63%. The random component accounts for 29.76% of the total variation in rank finishes.

The four-step procedure outlined in task one above was repeated for each of the 2012–19 F1 seasons individually. Table 7 reports these estimates based on the fixed effects FGLS regression results for the team and driver regression models.[27] The results suggest that the skill of the drivers contributes the least to explaining the rank finishes,

[25]Although we speak in terms of an increase in salary, we interpret it to reflect the driver's expected marginal product.
[26]The estimated coefficients for *racestarts* and *racestarts*^{2} suggest that a driver does not reach a maximum effect for race experience until approximately 75 prior races. After that point, further racing experience *reduces* the average rank finish.
[27]The coefficient estimates and summary tests for each of the F1 seasons is available in a working paper version upon request.

ranging from 9.88% in 2015 to 20.07% in 2017. In all F1 seasons, the team contribution accounts for a larger share of the variation in rank finishes, ranging from 14.85% in 2013 to 28.49% in 2018. Averaging across the 2012–19 F1 seasons gives values of 13.73% and 20.66% for drivers and teams respectively. The interaction of driver and team accounts for the largest share of the variation in rank finishes in each season, ranging from 28.41% in 2012 to 47.55% in 2013, and averaging 33.75% over all eight seasons. The intuition of this shared variation is similar to an interaction term in a regression model. The performance of a team is enhanced by a better-quality driver and vice-versa. The largest effect occurs when a top driver is placed on a top team and the smallest effect occurs when a lesser driver is placed on a lesser team. Generally, the teams with the largest budgets also hire the highest paid drivers (Mercedes, Ferrari and Red Bull in particular). The unexplained variation in rank finish account for between 22.14% (2018) and 34.99% (2012). These are random factors, outside of retirements due to crashes and mechanical issues, that affect the final order of finish.[28]

VII. Rule changes and complementarities

The 2012–19 seasons contained several seasons that marked significant rule changes to how the cars are constructed and their technical limits. Rule changes are typically determined by the FIA to promote greater parity in team budgets and the availability of technologies, as well as to increase driver and fan safety. Rule changes enacted in the 2012 season were designed to improve parity on the track and make driving safer. These changes required drivers to quickly adapt to new race strategies (passing rules, yellow flags, cornering lines, etc.), however only modest technical changes were made to the cars. Rule changes for the 2014 season also reflected the FIA's concern for environmental responsibility by requiring all cars to use turbo-hybrid engines (power units) that utilized two electric motors in addition to their smaller 1.6 litre V-6 gasoline engines.[29] The result was a far more fuel-efficient power unit that produced fewer exhaust emissions. Other changes for 2014 included the elimination of some performance-enhancing air effects to make the cars safer (lowering the front nose, eliminating side diffusers, etc.). Smaller teams struggled with the move to these expensive technologies and took several seasons to competitively adapt

Table 7 reveals that the variation in rank finishes explained by driver skill reached 16.7% in the 2012 F1 season, perhaps reflecting the rule changes that emphasized greater driving skill and strategy on the track. The interaction plus team shares of the variation totalled 48.31%, the lowest value for the 2012–19 sample period. The significant rule changes to the cars in the 2014 season corresponded with a decrease in the driver share of the total variation to just 11.52%, while the team share

Table 7. Variation in rank finishes explained by driver, team, interaction and unexplained. 2012–2019 F1 seasons.

Season	Sample size	*SST*	R^2(step 1)	R^2_D	R^2_{TD}	R^2_T	$1 - R^2_D - R^2_{TD} - R^2_T$
2019	420	13,965	0.5719	0.1264	0.3956	0.1763	0.3017
2018	420	13,965	0.6109	0.1677	0.3260	0.2849	0.2214
2017	400	13,300	0.5160	0.2007	0.2855	0.2305	0.2833
2016	462	18,595	0.6000	0.1206	0.4097	0.1903	0.2794
2015	380	12,887	0.6196	0.0988	0.4228	0.1968	0.2816
2014	352	14,193	0.6210	0.1152	0.3948	0.2262	0.2638
2013	418	16,954	0.6240	0.1021	0.4755	0.1485	0.2739
2012	480	22,964	0.4831	0.1670	0.2841	0.1990	0.3499

[28] A word of warning is appropriate here. We are not estimating a production function for F1 racing. Although we discuss complementarity of inputs, the data at hand are not adequate to provide a standard functional characterization since generally only two drivers characterize a team each season. Consequently, writing output as a function of team and driver leads to collinearity as the driver is the team. More subtle characterization of the inputs is needed to develop a true production function.

[29] The Kinetic Energy Recovery System (KERS) that charges electric batteries in the cars using braking was first required in the 2009 F1 season. However, fuel consumption was poor at 194 kg/hour due to the much larger V-10 engines. The 2014 turbo-hybrid system consumed only 100 kg/hour, allowing for a smaller, lighter fuel tank.

increased to 22.62%. These percentages are significantly different from their 2012 values based on the method in Olkin and Finn (1995).[30] As lesser teams adapted to the new rules in 2015, the driver share of variation dropped further to a low of 9.88%,[31] while the interaction plus team shares increased to 62.1%.[32]

Regardless of the rule changes, the driver share of the total variation in rank finishes is consistently the smallest share in Table 7. The interaction between driver and team consistently accounts for the largest share of variation, excepting the 2012 season in which the unexplained share is the largest. By itself, this shared variation between driver salary and team budget (not including driver salaries) does not imply these are complementary inputs since they could be negatively associated with each other, even though the association is strong. A negative association suggests these inputs are substitutes, which could certainly be the case if total team spending (budgets and salaries) is limited. The correlation coefficient between driver salaries and team budgets ranges between 0.574 and 0.799 in our sample. Better drivers and better teams appear to be significantly complementary inputs into the production function that produces rank finishes. Drivers do not just drive the cars but also provide valuable input and feedback on the development of the cars. Their labour is a sort of endogenous growth process that improves the technology of capital. This then feeds back into the productivity of the driver. Lesser drivers and teams do not experience this endogenous process to as great a degree.

VIII. Summary

F1 racing could easily be the most capital intensive and technologically advanced 'sport' in the world. Highly skilled drivers compete in complex racing machines that are difficult to master. This paper asks two questions: What are the shares of racing results attributable to driver skill and team technology? How should teams invest their scarce budgets in these two inputs? The simple 80–20 rule is found to be an over-simplification of the shares. Our regression results for the 2012–19 F1 seasons suggest driver skill and team technology uniquely contribute roughly 15% and 20% respectively to race outcomes (rank finishes), but that the interaction between the two complementary inputs accounts for between 30% and 40%. More skilled drivers improve the return to team technology and vice-versa. After all, F1 cars do not drive themselves and drivers cannot ply their trade without an F1 car. The random share of race outcomes is significant at 20–35%. Perhaps F1 world champion Nico Rosberg can be excused for ignoring the random component in his casual assessment of the shares, however ratio-scaling the shares still amounts to a roughly 22-30-50% split between driver, team and driver-team interaction. To say that drivers contribute only 20% is a vast underestimate given the critical complementarity between driver and team. Drivers do not just driver cars, but also provide valuable input into car development and testing. Our results broadly agree with Bell et al (2016) who utilizes what is essentially a two-factor ANOVA approach.

Where F1 teams best invest their scarce financial resources can only be determined by incorporating team budgets and driver salaries into our regression models. We allowed for an interaction effect, essentially a shift in the marginal product of the driver skill (team budget) when the team budget (driver salary) is increased. Although teams must spend large amounts to field even minimally competitive cars in F1, our results suggest that the return to hiring more driving skill (at an assumedly higher driver salary) is positive but diminishing in the size of the team budget. The return to spending more on the team budget is positive but diminishing in the size of the driver salary (assumedly driver skill). The upshot is that teams that spend more

[30]The 2012 and 2014 driver's shares are significantly different at 90% confidence. The team shares for 2012 and 2014 are not significantly different at any reasonable level of confidence, however the results are suggestive. A useful calculator for confidence intervals can be found at https://ptenklooster.nl/confidence-interval-calculators/confidence-intervals-for-r-square/.
[31]Significantly different from the 2012 value at 95% confidence.
[32]Significantly different from the 2012 value at 95% confidence.

team budgets and driver salaries will improve their rank finishes, but at a diminishing rate. This suggests an interesting maximization problem for a representative F1 team that we leave for future research (and more data).

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Aversa, P., P. Santi, and S. Haefliger. 2015. "Business Model Configurations and Performance: A Qualitative Comparative Analysis in Formula One Racing: 2005-2013." *Industrial and Corporate Change* 24 (3): 655–676. doi:10.1093/icc/dtv012.

Barzel, Y. 1972. "The Rate of Technical Progress: The 'Indianapolis 500'." *Journal of Economic Theory* 4 (1): 72–81.

Bell, A., J. Smith, C. Sabel, and K. Jones. 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. doi:10.1515/jqas-2015-0050.

Boll, R. 2020. "How to Win in Formula One: Is It the Driver or the Car?." *The Correspondent*. Accessed 30 June 2021. https://thecorrespondent.com/642/how-to-win-in-formula-one-is-it-the-driver-or-the-car/84931340736-b6c94330

Budzinski, O., and A. Muller-Kock. 2018. "Is the Revenue Allocation Scheme of Formula One Motor Racing a Case for European Competition Policy?" *Contemporary Economic Policy* 36 (1): 215–233. doi:10.1111/coep.12247.

Davidson, R. 2009. "Reliable Inference for the Gini Index." *Journal of Econometrics* 150 (1): 30–40. doi:10.1016/j.jeconom.2008.11.004.

Eichenberger, R., and D. Stadelmann. 2009. "Who is the Best Formula One Driver? an Economic Approach to Evaluating Talent." *Economic Analysis and Policy* 30 (3): 389–406. doi:10.1016/S0313-5926(09)50035-5.

Judde, C., R. Booth, and R. Brooks. 2013. "Second Place is First of the Losers: An Analysis of Competitive Balance in Formula One." *Journal of Sports Economics* 14 (4): 411–439. doi:10.1177/1527002513496009.

Lazear, E., and S. Rosen. 1981. "Rank-Order Tournaments as Optimum Labor Contracts." *The Journal of Political Economy* 89 (5): 841–864. doi:10.1086/261010.

Mantel, S., G. Rosseger, and S. Mantel. 1995. "Managing Technology at the Indianapolis 500." *Technological Forecasting and Social Change* 48 (1): 59–76. doi:10.1016/0040-1625(94)00033-S.

Marino, A., P. Aversa, L. Mesquita, and J. Anand. 2015. "Driving Performance via Exploration in Changing Environments: Evidence from Formula One Racing." *Organization Science* 26 (4): 1079–1100. doi:10.1287/orsc.2015.0984.

Mastromarco, C., and M. Runkel. 2009. "Rule Changes and Competitive Balance in Formula One Motor Racing." *Applied Economics* 41 (22–24): 3003–3014. doi:10.1080/00036840701349182.

Nalebuff, B., and J. Stiglitz. 1983. "Prizes and Incentives: An Economic Approach to Influence Activities in Organizations." *Bell Journal of Economics* 14 (1): 21–43. doi:10.2307/3003535.

Olkin, I., and J. Finn. 1995. "Correlations Redux." *Psychology Bulletin* 118 (1): 155–164. doi:10.1037/0033-2909.118.1.155.

Pesaran, M. 2021. "General Diagnostic Tests for Cross-Sectional Dependence in Panels." *Empirical Economics* 60 (1): 13–50. doi:10.1007/s00181-020-01875-7.

Phillips, A. 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.

Sarafidis, V., and T. Wansbeek. 2012. "Cross-Sectional Dependence in Panel Data Analysis." *Econometric Reviews* 31 (5): 483–531.

Schreyer, D., and B. Torgler. 2018. "On the Role of Race Outcome Uncertainty in the TV Demand for Formula One Grands Prix." *Journal of Sports Economics* 19 (2): 211–229. doi:10.1177/1527002515626223.

Sylt, C. 2015. "European Commission Brushes off Formula 1 Antitrust Allegations." *Forbes Magazine*. Accessed 19 May 2022. https://www.forbes.com/sites/csylt/2015/01/06/european-commission-brushes-off-f1-antitrust-allegations/?sh=79d96e1c6b5a

Wesselbaum, D., and P. Owen. 2021. "The Value of Pole Position in Formula One History." *Australian Economic Review* 54 (1): 164–173. doi:10.1111/1467-8462.12401.

APPENDIX

Data sources:

Variable	Definition	Source
rankfinish	Race position at end of race	https://www.f1-fansite.com/f1-results
teamexp	Total team expenditures excluding driver salaries	https://www.spotrac.com/formula1
salary	Driver salary in real US Dollars	https://www.spotrac.com/formula1
poll	Poll position for each driver at start of race	https://www.f1-fansite.com/f1-results
devpittime	Deviation of fastest pit stop time for team from the F1 season average fastest pit stop time for all teams	https://www.motorlat.com/and author's calculations
racestarts	Number of F1 races started prior to the current race	https://www.f1-fansite.com/f1-results and author's calculations
teamdnf	*teamdnf* = 1 if car did not finish race due to team fault	https://www.f1-fansite.com/f1-results
driverdnf	*driverdnf* = 1 if car did not finish race due to driver fault	https://www.f1-fansite.com/f1-results