POLITECNICO DI TORINO

Master's of Data Science and Engineering

Master's Degree Thesis

Data-Driven Vehicle Performance Optimization for Formula Student Racing

Supervisors

Prof. Andrea TONOLI

PhD Candidate Stefano FAVELLI

Dario SALZA

Federico OLDANI

Candidate

Lal AKIN

October 2024

Aileme

Abstract

There is only one goal to achieve in the world of motorsports; to cross the finishing line before everyone else. However being the fastest one is not as simple as pushing the throttle and driving in circles. There are numerous factors influencing the performance of a race car, from how the driver handles the vehicle to the unpredictable environmental factors surrounding the track and the vehicle.

Formula Student Racing, where student-formed teams are required to build and race vehicles, is a university level single-seater racing competition. It's a hands-on learning experience where engineering students combine their theoretical knowledge with real-world challenges and push the boundaries of automotive technology.

Building a fast vehicle is only part of the equation for Squadra Corse PoliTo, the Formula Student racing team of Politecnico di Torino. The real challenge is the integration and optimization of the driver's and vehicle's performance. This challenge is the the main focal point of this thesis. Data Science and Machine Learning methods are implemented to understand the impact of the driver's actions on the vehicle, and to maximize the overall performance.

The goal is to develop diverse yet interconnected tools for Squadra Corse PoliTo to offer insights about the underlying relationships between the driver input actions and vehicle responses. By deploying the power of data analysis and machine learning, the purpose is to provide engineers and drivers of Squadra Corse PoliTo with the competitive edge and deeper understanding they need to outpace themselves and the competition.

First, critical inputs of driver behaviour on vehicle performance will be identified. Then, following this knowledge predictive models that can anticipate how different strategies will impact performance will be constructed for diverse performance metrics. Final tool will be built by analysing the driving styles of the drivers, extracting expert demonstrations to create a driver model that has the ability of making informed decisions as a driver would. With this proposed framework, Squadra Corse PoliTo can formulate the best strategy configurations for specific scenarios and give personalized feedback to the drivers to maximize performance in Formula Student competitions.

Keywords:Formula Student Racing, Machine Learning, Outcome Prediction, Strategy Optimization, Driving Style, Gaussian Process, Neural Networks, Imitation Learning, Driver Models

Acknowledgements

Firstly I would like to express my immense gratitude to Prof. Andrea Tonoli for his guidance, interest and support during the process of researching and writing this thesis.

I am beyond grateful to Stefano Favelli for his dedication to improving this work and enthusiasm for exchanging endless ideas. Thank you for the time and effort you have put in, your belief in my abilities has been a constant source of motivation.

I would like to convey my gratitude to Dario Salza and Federico Oldani for sharing their invaluable expertise with me. Their wise guidance during moments of uncertainty were instrumental in the completion of this work.

I wish to express my gratitude to my colleague, Gabriele Roccatani, for his belief and interest in my ideas. If he had not given my proposal a chance and spent countless hours answering my questions I wouldn't have been able to complete my thesis.

Special thanks are due to Squadra Corse PoliTo for their interest and involvement, which enabled me to pursue this research project. Thank you for allowing me to work with you, it has been a pleasure. I am also grateful to LINKS Foundation for providing me with an environment for my academic research.

My dearest friends, I owe you the hours I've stolen from you by talking your ears off about my ideas, doubts and excitements about my thesis. I am so lucky to have people in my life such as you, who celebrate my accomplishments as if they are theirs.

I would like to thank my support system, for there is no doubt about your place in my heart. Your encouragement and love have been my rock. Thank you for believing in me when I didn't.

Finally, I am forever and endlessly indebted to my family for their love and unwavering support. Nothing I have accomplished so far would've been possible without the sacrifice and hard work of my dear parents. Their dedication has been the foundation for everything I am today, and everything I can ever dream of being.

Table of Contents

Chapter 1

Introduction

"Thirty-five years later, I can look back on an eventful, fruitful career, one spent designing cars and asking myself the same series of simple questions. How can we increase performance? How can we improve efficiency? How can we do this differently? ***How can I do this better?"***
-How to Build a Car, Adrian Newey

Since the first car was built in the 19th century, racing has been an integral part of its existence. Human beings have always been captivated by the idea of designing the fastest car, the one that can beat all others, and reach the chequered flag the earliest. Perhaps to satisfy this goal, people from all around the world are following competitive vehicle racing, also called Motorsports.

There are many different divisions in Motorsports, such as Formula 1, Formula E, NASCAR, Indycar... Formula 1 is arguably the most well-known and followed of them all, amassing a massive number of new spectators every year. The ever-rising fame of Formula 1 has given way to many research possibilities in the field of motorsports, and topics such as Race Outcome Prediction, Strategy Maximization, Driver Modeling and Lap-time Simulation are attracting researchers interest. These new areas of research are developed by using novel Machine Learning, Data Science and Reinforcement Learning techniques. Designing the fastest car, foreseeing race outcomes and preparing beforehand, giving the driver the most critical strategy calls based on projections are some of the usages of these techniques. For motorsports teams, the advancement in technology means being able to create many different scenarios of a race before it even happens, and testing the car with different configurations without the driver having to test all of them physically.

There are many different levels of Formula Racing, one of them being Formula Student Racing, which is a university-level competition designed to push young engineers to design, build, and race their own vehicle each year.

Squadra Corse PoliTo is the Formula Student Team of Politecnico di Torino, and

has been a powerful player in the Formula Student Electric Vehicle competitions for many years.

This thesis work is done in collaboration with Squadra Corse PoliTo, in order to;

- Reveal invaluable insights about relationships between driver inputs and vehicle outputs
- Leverage the relationships to predict vehicle performance
- Analyse driving style indicators to reveal to extract "expert demonstration" data taken from the better performing styles
- Optimize driver input and driving style in order to achieve maximum performance and to give feedback to drivers on how to improve their results
- Create a driver model that can make strategic decisions based on expert demonstrations, that provides an easier alternative to testing different strategies and configurations

1.1 Formula Student Racing

Formula Student is a university-level vehicle engineering competition that has been organized for students worldwide by SAE International (Society of Automotive Engineering). The first edition of Formula Student took place in the USA, in 1981. During the early 2000's, the competition made its way to Europe, and in 2005 Italy joined the list of countries that has hosted the races. Today, over 400 university teams from more than 60 countries take part in the competition as shown in 1.1 [1].

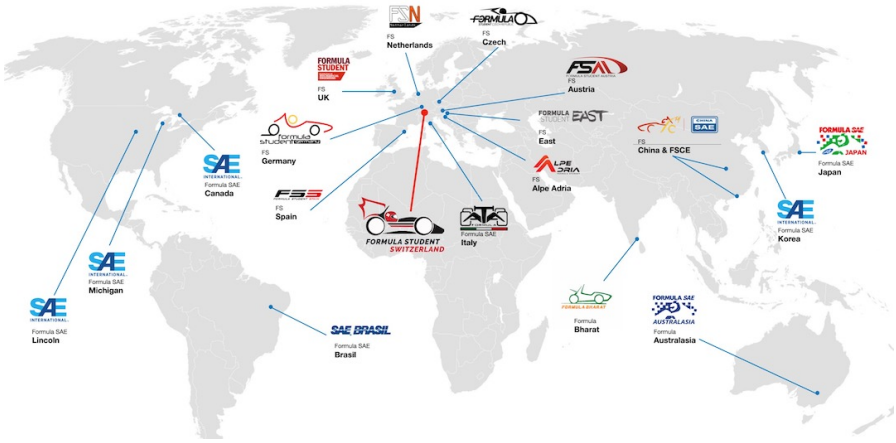

Figure 1.1: Formula Student events.

The teams are divided in 3, namely; Combustion Engine Vehicles, Electric Vehicles, and Driverless Vehicles. Electric Vehicle competition has become the most competitive and advanced type because of the current rise in popularity of EV's and also the novel engineering challenges it offers compared to combustion engine cars.

Each of the vehicles produced by teams can only be used for one year, which means the car must be changed significantly every year. Following this requirement, there is always ongoing production, testing and designing processes for the teams. Also the rules prohibit any direct involvement of professional engineers or mechanics to challenge engineering students even further.

Figure 1.2: Formula Student Austria Competition, 2024.

Since Formula Student events are not inherently racing championships but more so engineering competitions, there are diverse areas for the teams to compete in. Each Formula Student event is divided in two main categories; Dynamic and Static Events. Each event adds points to the team's overall score, and the diversity of types of evaluation criteria ensure awarding every aspect of engineering.

- Static events:
  - Engineering Design Event: Teams present the design of the vehicle to a panel of judges, who assesses its innovation, feasibility, and engineering. The judges may ask questions to test the team's knowledge of the vehicle.
  - Cost and Manufacturing: For this event the teams must present their cost plan for manufacturing and production of the car. It should be detailed and factual, encompassing all costs for the current year and manufacturing plan and steps.

  - Business Plan: Teams must present a business plan containing their strategies for marketing and sponsorship and the management of financial activities.

- Dynamic events:
  - Acceleration: This event is preformed on a 75 meter straight-line. The fastest car wins this step that is dependent mostly on how fast the car will reach high speeds.
  - Skidpad: The car is required to do 4 laps, 2 on each side, on a figure of eight. Second lap of each side is timed to measure the vehicle's stability and ability to take tight turns.
  - Autocross: This event is the most similar to a qualifying lap for Formula 1. The car is required to go through different corner types and chicanes, testing the vehicle itself and the skill of the driver.
  - Endurance and Efficiency: The longest event among the previous ones, this step evaluates the vehicle's reliability. For this event, a 22 km long track with different corners is required to be completed. The energy consumption is calculated as a performance indicator amongst others.

1.2 Squadra Corse PoliTo

Squadra Corse PoliTo is the Formula Student team of Politecnico di Torino. When Squadra Corse was founded in 2004, the first vehicle made for the 2005 season was an internal combustion engine vehicle.

After participating in the hybrid class in 2010 and winning the championship, SCP constructed their first electric vehicle at 2012. This advancement earned

Figure 1.3: Evolution of earlier Squadra Corse PoliTo cars.

Squadra Corse PoliTo the title of being the first Italian team to participate in EV-class. After continuing to develop rapidly and gaining experience, SCP achieved one of their best performances at Formula Student Italy 2019 which is held at Varano de' Melegari track. The team won gold medal in this competition particularly excelling in Acceleration and Endurance events.

Figure 1.4: Evolution of EV era Squadra Corse PoliTo cars.

To this day, Squadra Corse races under division class 1EV (for electric vehicles), continuing on their path of carrying automotive engineering to the future and developing sustainable solutions for the future.

Squadra Corse PoliTo has diverse divisions to handle the complicated engineering challenges they are faced with each year:

- Management
- Aerodynamics and CFD
- Chassis and Compsites
- Powertrain
- Unsprung Masses and Geartrain
- Electronics
- Vehicle Dynamics and Control System
- Battery Pack
- Communication and Media
- Thermal Management

From August of the previous year the engineers of SCP set their targets and goals for the next season. September to January is the design phase of the new vehicle, followed by Production and Assembly phases that last until June. After this, the vehicle is prepared and tested on track during the whole summer. August is the month in which the Formula Student events take place, and following this, the whole process begins again. This thesis followed SCP throughout the 23/24 season.

Feature	Specification
Monocoque	Entirely made of carbon fiber
Total Weight	200 kg
Max Power	80 kW (limited by rules)
Aerodynamics	Cz = 4.8 | Efficiency = 3.1
0 – 100 km/h	2.6 seconds
Max Speed	122 km/h
Dimensions	293 x 140 x 120 cm
Traction	4 WD
Motors	4x AMK electric motors

Table 1.1: Technical Specifications of the SCP Vehicle

1.3 Motivation

By implementing state-of-the-art tools and approaches, this project seeks to address the key challenges faced by Squadra Corse PoliTo and contribute to the broader field of automotive engineering.

The field of automotive engineering is evolving rapidly, driven by the need for more efficient, sustainable, and high-performance vehicles. After conducting extensive research on this domain, it is seen that there exists a gap concerning the amount of existing research for Formula 1 compared to Formula Student Racing. The lack of research and development for Formula Student Racing is the primary motivation behind this thesis work.

Despite the progress made in recent years, optimizing a Formula Student car for maximum performance and efficiency remains a complex challenge. Teams must balance a wide range of factors, including basic safety regulations, aerodynamics, following competition rules, vehicle and software design. To remedy some of these challenges, achieving a deeper understanding of the nature of the interactions between driver and vehicle is chosen as the focus of the proposed thesis. The work proposed will be towards increasing the team's performance in the Dynamic Events

Figure 1.5: Squadra Corse PoliTo 2024 vehicle, Andromeda.

of a FS event, given that the driver strategy and vehicle performance are crucial elements for these events since in the Autocross and Endurance rounds, driver skill and strategy; vehicle performance and balance reveal the winner. The general framework of the research is given in Figure 1.6.

To achieve this, the following research questions will be addressed, and respective tools will be developed:

Question 1: How can the deeply complex relationships between driver actions and vehicle reactions be mathematically analyzed?

Tool 1: Correlation Analysis and Descriptive Statistics to discover the nature of features followed by Feature Selection to improve the quality and explainability of the models.

Question 2: How can the results from Tool 1 be expressed in a mathematical model, in which the vehicle performance outputs are used as the dependent variables, and driver inputs as the independent variables?

Tool 2: Using different Regression models to predict vehicle performance output variables, and reverse engineering to derive the function that links the variables together.

Questions 3, 4, 5: Is there a distinct driving style for every driver, even if

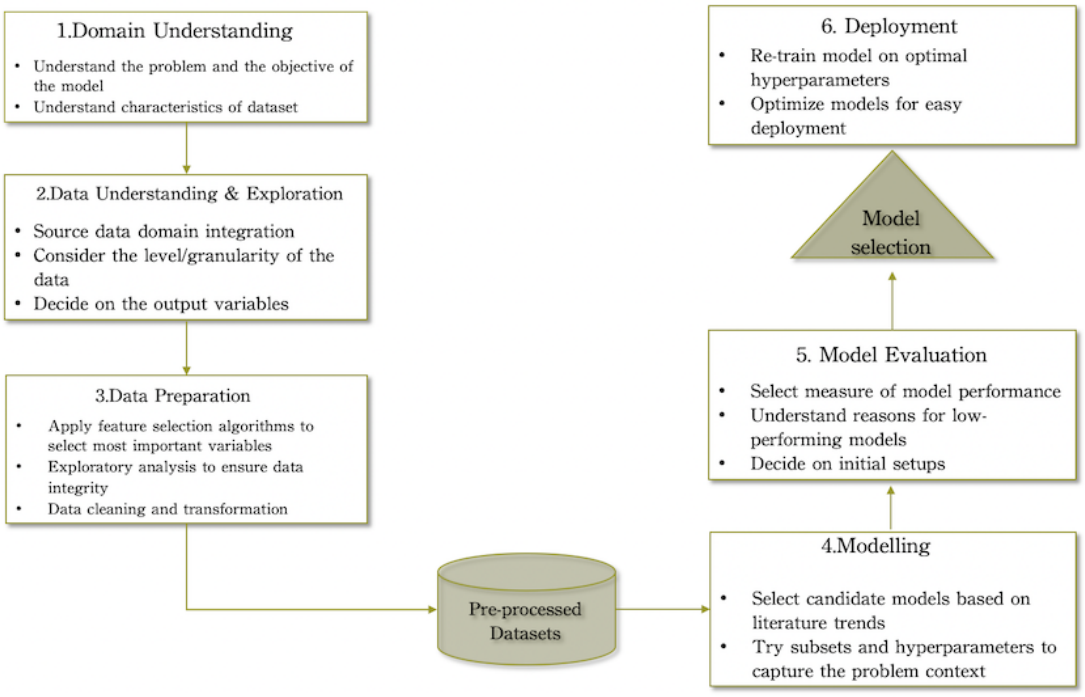

Figure 1.6: Framework for Data-Driven Performance Optimization analysis.

they are driving the same car on the same track? How can human behaviour be measured and quantified? Will the driving styles be distinct enough to classify by Machine Learning models?

Tool 3: Using visualization tools to determine specificity in driving style and to measure the performance of drivers. Using these insights to build classification models to identify each driver by their driving style.

Question 6: How can the vehicle's performance be optimized by adjusting driver inputs in test simulations or feedback sessions for drivers?

Tool 4: Numerical Optimization methods and Driving Style Analysis to extract best-performing actions taken for each corner and building feedback on these findings, aiming to maximize performance.

Question 7: Can AI models learn and replicate expert drivers' strategies and behaviors to create a driver model, which can substitute human drivers for simulator tests?

Tool 5: Imitation Learning techniques, such as Behavior Cloning, to model expert drivers' decisions and replicate their driving styles under varying states.

By addressing these questions, this research aims to develop a comprehensive optimization framework for Formula Student vehicle of SCP and the drivers,

resulting in enhanced performance, efficiency, and competitiveness. The findings of this thesis will benefit the team by providing valuable insights, driving further the innovation and development.

1.4 Thesis Outline

This thesis work is structured in 7 chapters and the outline is as follows:

1. **Chapter 2 Theoretical Background** provides an in-depth exploration of the theoretical concepts related to Data Science, with a particular focus on Machine Learning and Optimization applications in motorsports.
2. **Chapter 3 Data Collection and Analysis** explains the methodology for collecting data and different datasets, followed by a detailed explanatory data analysis step.
3. **Chapter 4 Outcome Prediction** discusses various Machine Learning models and approaches used to predict outcomes based on performance data and vehicle dynamics.
4. **Chapter 5 Driving Style Analysis** presents an analysis of different driving styles, examining how these styles influence vehicle performance and overall results.
5. **Chapter 6 Optimization and Imitation Learning** explores Numerical Optimization and Imitation Learning, focusing on their possible use in performance maximization and strategic decision-making.
6. **Chapter 7 Conclusions** summarizes the work done in the thesis, presents the key findings, and outlines potential directions for future steps.

Chapter 2

Theoretical Background

In this chapter, a literature review will be given to provide an overview of the current state of research in the domain of Machine Learning applications to motorsports. To accomplish this, topics that are necessary to form a detailed and whole understanding such as Vehicle Dynamics, Driving Style, Machine Learning Theory and Methodology will be explained; followed by a survey of the state-of-the-art methods of this domain.

The survey of the current usages and state-of-the-art methods of Machine Learning for motorsports will include outcome prediction, strategy optimization and driver modelling and highlight the innovative potential of these technologies in carrying competitive racing to the future.

Throughout this literature review, the key challenges and potential solutions in Formula Student performance optimization will be identified, providing a solid foundation for the research questions and methodologies outlined in this thesis.

2.1 Motorsports

The term "motorsports" encompasses different levels of competitive automobile racing disciplines from karting to more advanced levels such as Formula 1 Racing. Fundamental goal is the same across the different disciplines: optimizing vehicle performance and maximizing driver skill to achieve the fastest time. To achieve this goal, teams leverage Data Analysis and Machine Learning given that the amount of data that can be collected from motorsport events is growing exponentially and the teams must be able to handle this trend. This fast-growing trend creates a demand for technological advancement in the field of motorsports. [2]

In top-level motorsports, drivers must collaborate effectively with their race engineers to fine-tune vehicle setups to their driving style and specifics of the track,

and this dependency further blurs the line between human input and mechanical optimization since one without the other is not sufficient.

> "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. -Duane Rockerbie" [3]

In recent years, research in motorsports has increasingly focused on data-driven approaches to improve both driver and vehicle performance rather than simply predicting race results. The rise of ML and AI technologies has further fueled research into motorsports, with techniques like Reinforcement Learning, Imitation Learning (IL) and Lap-time Simulation. [4] [5]

To understand the absolute necessity of using high-level technologies in the field of motorsports, a detailed explanation into the two factors, driver and vehicle, must be made. It is inherently impossible to split the two factors, and to look at them individually. The driver interacts with the vehicle by the control actions (steering, braking, accelerating), but these actions are highly influenced by the previous reactions of the car. Therefore driver skill and vehicle performance are two halves of a whole, which is overall performance. The main contextual factors of vehicle dynamics and current research into driving style and its effects will bring the nature of this co-existing relationship to light.

2.1.1 Vehicle Dynamics

Discussing the vehicle dynamics aspects in motorsports is crucial for understanding how the vehicle performs on the track and how it responds to driver inputs. Key forces such as yaw, grip, understeer, and slip significantly influence the car's speed, handling, and overall lap time. [6] [7]

Another very important factor on vehicle performance is tire wear, which is caused by the friction between the track surface and the tire. The friction creates thermal wear on the tire surface which means the thinning of the surface, and it directly affects grip, vehicle handling, and overall performance, particularly during cornering. Bringing and keeping the tires in the right temperature window allows efficient cornering maneuvers, provides maximum grip, while both overheating and underheating can lead to reduced grip and, consequently, slower corner times. Developing an understanding of how to use the tires is a key skill each driver should master. [8]

- *Yaw* refers to the rotational movement of the vehicle around its vertical axis, determining how the car turns [9]. Efficient control of yaw allows drivers to navigate corners smoothly and maintain higher speeds through turns.

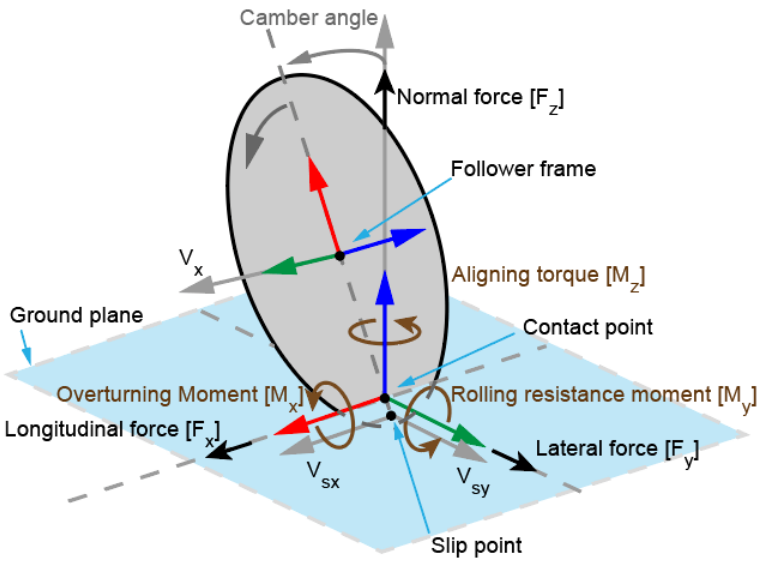

Figure 2.1: Forces acting on a tire.

Excessive yaw or poor control can lead to instability, increasing lap times due to slower speeds.

- *Grip* is the coefficient of friction between the tires and the track, directly affecting acceleration, braking, and cornering. High grip enables the car to transfer more power to the road, improving both speed and stability. Loss of grip, however, leads to slip and can cause the vehicle to perform poorly, particularly in corners.

- *Understeer* occurs when the front tires lose grip during a turn, causing the front of the vehicle to turn less than the rears. This forces the driver to slow down to regain control, impacting lap times negatively.

- *Slip* refers to the condition when the tires exceed their optimal grip and begin to lose contact with the road. Tire slip can occur during both acceleration and braking and is damaging to the control and balance of the vehicle.

A G-G plot, or traction circle is a graphical representation used to analyze the forces acting on a vehicle's tires during cornering, acceleration, and braking [10].

In this graphical representation, lateral and longitudinal acceleration is mapped in order to see how the vehicle handles steering, acceleration, and braking. As it is explained in Figure 2.2, the outer edges of the circle represents the peak grip that the tires can generate. Traction circle of a lap can give insights as to how much of the potential is being used by the driver. When a vehicle has a trace nearing the

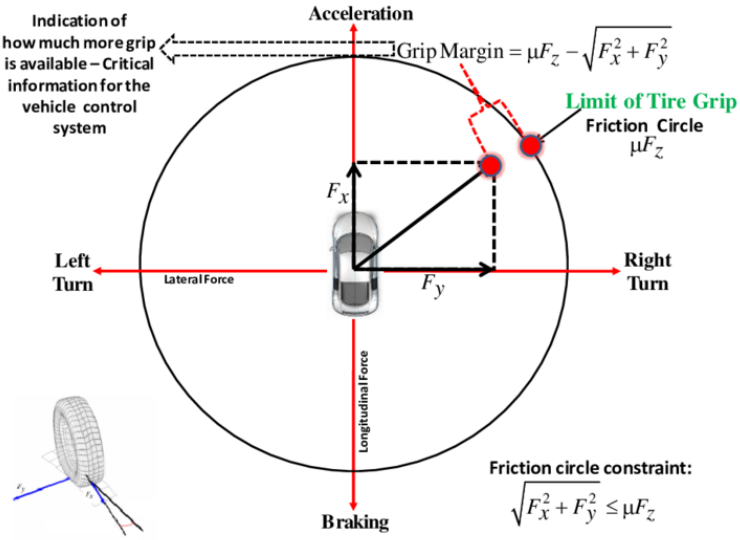

Figure 2.2: Traction circle details by Singh et al.

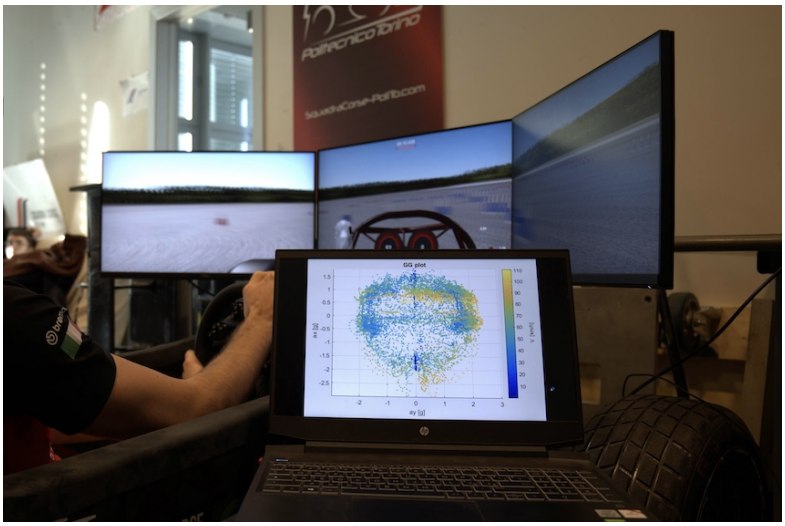

Figure 2.3: Traction circle of a simulator lap in SCP Office.

outer edges, it can be said the driver is skilled at tire management. Understanding these relationships between driver skill and vehicle performance through variables present in the dataset is crucial to optimize performance.

2.1.2 Driving Style

What makes a driver skilled in handling the racing vehicle is whether or not he/she can achieve the delicate balance between pushing the vehicle to its limits to achieve the fastest speed and ensuring the stability. Main three actions of a driver to control the vehicle are pushing the throttle pedal, pushing the gas pedal and steering the wheel. These three actions result in acceleration, deceleration or turning, respectively. The driver must handle all of these actions in a harmonious way in order to extract the highest level of performance. As a consequence, understanding and mastering the aforementioned vehicle dynamics phenomena is of the utmost importance.

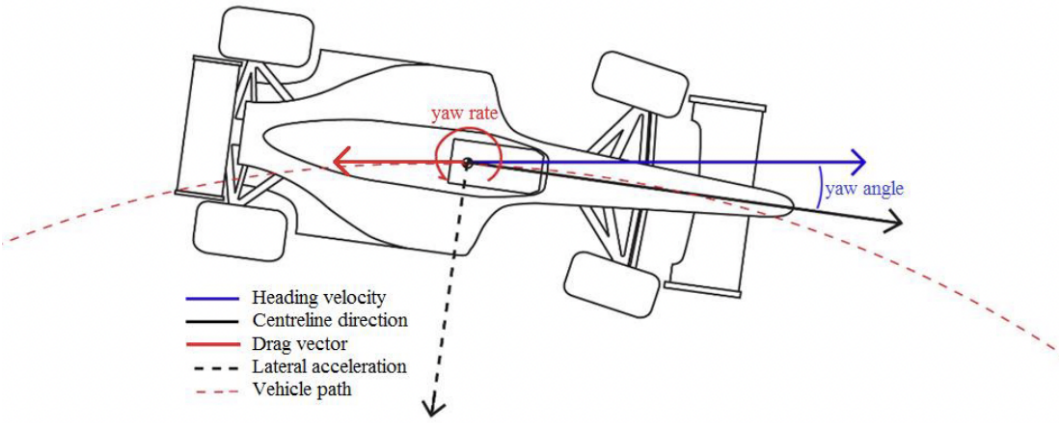

Figure 2.4: Forces acting on a Formula racing vehicle.

Driver skill plays a critical role in managing the said dynamics of driving. However, quantifying driving skill is not straightforward, given that drivers are human beings before all else and have a degree of randomness in their actions. Each driver is different, having different patterns for braking, throttle pedal actuation and turning the steering wheel. [11] Experienced drivers are adept at sensing the car's behavior and adjusting their driving style to optimize the overall outcome.

For example, a skilled driver can apply throttle and braking precisely to maximize grip through a turn while avoiding excessive yaw or slip. Balanced behaviours as such help maintain higher average speeds throughout corners, ultimately increasing performance without sacrificing stability. What makes a good driver different than others is not a quantifiable factor and different indicators can put more importance to diverse skills. [12] However, the effects of the overall skill factor can be seen in the specific behaviours of the driver such as braking precision or smooth pedal actuation, cornering maneuvers and consistency.

In Formula Student racing, the driver's ability to control vehicle dynamics often sets top-performing teams aside from others, as the marginal gains from improved handling and speed management can result in a significant competitive advantage especially in Dynamic Events where the skill of driver is measured alongisde with the one of the vehicle.

2.1.3 Driver in the Loop Simulation

For explaining the components of Driver in the Loop (DIL) Simulators control system approach was used by Macadam et al.[13] Human driver is the control system that provides inputs such as throttle, steering and braking, into the plant, which is the vehicle. The plant then responds to these input actions and provides feedback in the form of sensory information, kick-back from the steering wheel, visual information and pedals, to the control system. This action and reaction loop is a direct copy of driving on a real track. [14] [15]

The most prominent advantage of a DIL Simulator is the controlled environment it allows to create, enabling teams to track all relevant variables in real time, hence enabling reproducible and consistent analysis.

Figure 2.5: Driving simulator of Squadra Corse PoliTo.

Driving simulators have been used by FS-level teams, either entirely built in-house [16] or partially outsourced like SCP, to increase the efficiency of testing different setups and designing the vehicle.

2.2 Machine Learning Theory and Methodology

Machine Learning methods are the basis for this thesis work. Various methods are utilized in this thesis coming from different families of ML methods such as

Regression, Classification and Imitation Learning. These methods are used to model the relationships between input features and output variables, to make predictions for unseen data and model drivers. In the following part a detailed review of Machine Learning methods will be presented, focusing also on the statistical tools used in the context of this thesis.

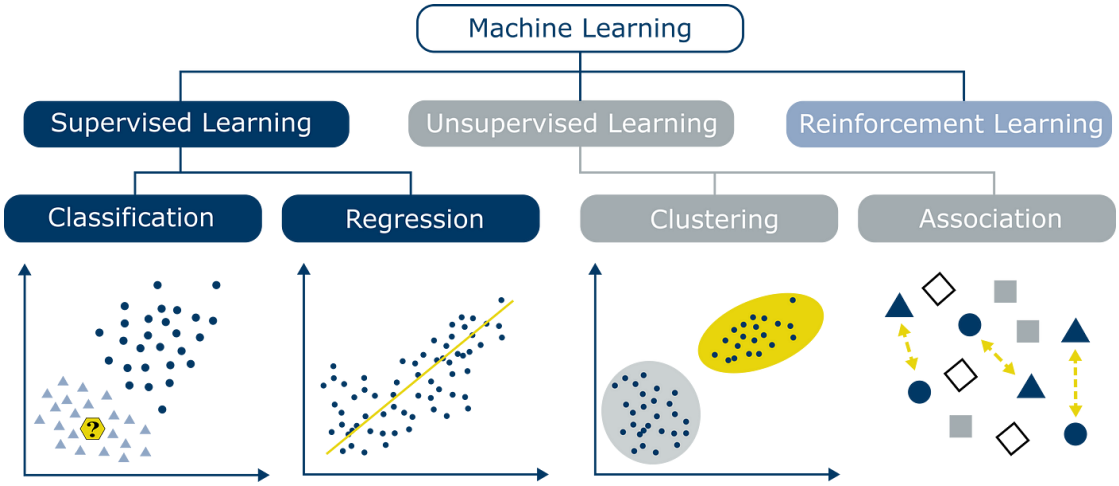

Figure 2.6: Machine Learning method families.

Supervised learning algorithms aim to learn a function from labeled data and make predictions by using this function [17]. In this research work, supervised learning algorithms, regression and classification are used. Unsupervised learning consist of methods such as clustering and learning-by-association, and the main difference is the lack of labeled data to learn from. Reinforcement learning is another family of ML algorithms, in which an agent is trained with loss and reward functions to learn a optimal policy [18]. A recently developed method called Imitation Learning is similar to RL in training a smart agent on decision making criteria, however for IL there is no reward or loss function, the expert data demonstrations are learned and imitated [19]. Imitation Learning is also used in this thesis as an alternative to Optimization for the strategy optimization task.

2.2.1 Supervised Learning

Regression Algorithms

Regression algorithms such as Linear Regression, Decision Tree Regression, Random Forest, Neural Networks, Ridge and Lasso Regression, Support Vector Regression, Logistic Regression and Gaussian Process Regression predict continuous

output variables from the labeled input data.

Linear Regression is one of the most well-known and straightforward ML techniques for making predictions. It is a simple yet powerful regression technique that finds a relationship between the independent variables X and the target variable y as a linear combination of the input variables $X_1, X_2, ..., X_n$:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon$$

Where $\beta_0, \beta_1, ..., \beta_n$ are the coefficients that the model learns from the training data and ϵ is the error term quantifying the numerical margin of error.

In the context of this research, linear regression is an important tool because it allows the mathematical relationship between the input features and the output variable to be explicitly known. Linear Regression is very limited in its applications since it finds linear relationships between variables and most use-cases involve non-linear datasets. This poses a trade-off between the explainability of results and complexity that the model can handle.

Decision Trees use a hierarchical tree-like structure to split the recursively data based on splitting criteria and aiming to reach a lead node to make a prediction. The splitting is made on the feature node that results in the highest error reduction. Decision Trees are easy to interpret like Linear Regression however they are prone to overfitting the data if the tree depth is not handled in a suitable manner.

Random Forest is an ensemble method that puts together many decision trees and then improves on the accuracy of the "forest" by averaging the predictions made by individual trees. Each tree is trained on a random subset of the dataset to reduce the possibility of overfitting.

Boosted Decision Tree is an ensemble learning method that improves the performance of decision tree models by combining the predictions of learners to form a strong overall model by training each tree sequentially. This method functions by each tree trying to correct the errors of the previous predictions.

Support Vector Regression is another version of the Support Vector Machine (SVM) algorithm that performs regression rather than classification [20]. SVR model finds a hyperplane, also called a decision boundary, that fits the data the best and minimizes the loss function The key difference of SVR with other regression algorithms is the usages of kernel functions. A kernel function allows projecting the input features into a higher-dimensional space. By using this mapping, the

SVR model is able to capture non-linear relationships better.

Gaussian Process Regression is a non-parametric, powerful and lesser-known regression algorithm that benefits from modelling the problem as a Gaussian Process. Unlike linear regression, GPR doesn't limit the space of relationships between inputs and outputs to a linear and fixed context but instead models the output as a "Multivariate Gaussian Distribution" of possible functions of the input variables [21]. GPR relies on a kernel function to map the relationship between the data points.

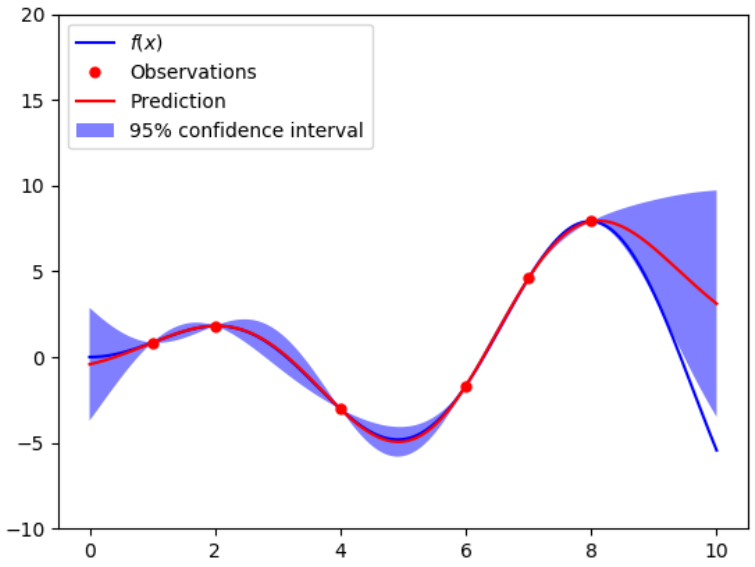

Figure 2.7: Visualization of the working principle of GPR by Subhasish et al.

Working principle of Gaussian Process Regression is to find the most suitable distribution for each variable out of the infinite distributions they can have. After fitting individual distributions to the input features, GPR performs regression based on the parameters of the set of distributions [22]. GPR was selected in this work for its capabilities in modelling each feature with the best-fitting individual distributions. GPR has numerous advantages such as performing well on small datasets and providing uncertainty measurements on each of the predictions.[23] In Galeazzi et al. [24] GPR was compared with Polynomial Regression and was found to perform better especially in fitting a distribution to each variable to increase explainability. It was noted also in their work that the performance of GPR was due to its complex and Gaussian nature.

GPR assumes the problem can be modelled as a Gaussian Process (GP) over the function $f(x)$, which is used to model the relationship between inputs X and outputs y.

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

A GP has a mean function $m(x)$ and a covariance function (kernel) $k(x, x')$.

Covariance Matrix (Kernel): The kernel function $k(x, x')$ stores the covariance between x and x'. Given n input data points, the covariance matrix $K(X, X')$ is constructed as:

$$K(X, X') = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix}$$

Joint Distribution of Training and Test Data: The joint distribution of the training outputs $\mathbf{y}$ and test outputs $\mathbf{f}_*$ (given inputs $\mathbf{X}$ and $\mathbf{X}_*$) is modelled as a Multivariate Gaussian Distribution:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

$\sigma^2 I$ is the noise in the data.

Posterior Distribution (Predictions): Given the training data, the posterior distribution of the test outputs $\mathbf{f}_*$ is Gaussian with mean μ_* and covariance Σ_*:

$$\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\mu_*, \Sigma_*)$$

The mean μ_* and covariance Σ_* of the posterior are given by:

$$\mu_* = K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}\mathbf{y}$$

$$\Sigma_* = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}K(X, X_*)$$

Log Marginal Likelihood: The log marginal likelihood is used to tune hyperparameters of the kernel function. It is defined as:

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T(K(X, X) + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K(X, X) + \sigma^2 I| - \frac{n}{2}\log 2\pi$$

Model	Equation	Terms
Linear Regression	$y = X\beta + \epsilon$	$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
GPR	$h(x)^T\beta + f(x)$	$P(y\|f, X) \sim \mathcal{N}(y\|H\beta + f, \sigma^2 I)$

Table 2.1: Comparison of Linear Regression and GPR

Neural Networks are powerful learning algorithms inspired by the human brain's structure, such as computational nodes represented as neurons as in Figure 2.8 by Wang et al. [21]. NN's have acquired a high level of acceptance in ML domain for their significant success in handling large and complex datasets.

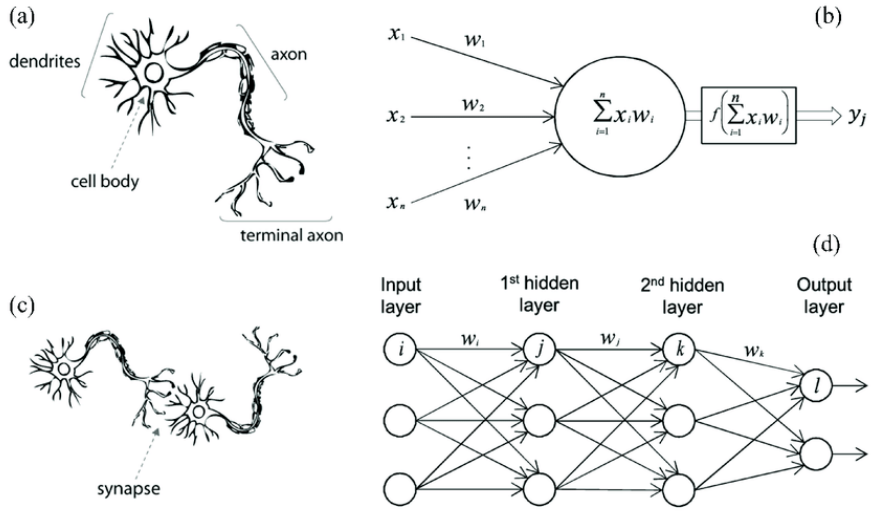

Figure 2.8: Comparison of neurons and Neural Networks.

General architecture of a Neural Network consists of an input layer, one or more hidden layers and an output layer. The layers are made up of interconnected neurons, or computational nodes, that work by minimizing a loss function by back-propagation and optimizing the findings at each pass. Each neuron applies a weighted sum of the information it receives, and outputs the result to the next neurons by activation functions. Neural Networks are also used in Classification tasks and the difference is determined by the output layer structure and activation functions that are used.

NN's are highly flexible and have many different modalities that can be applied to a various number of tasks. A downside of NN's is the possibility of overfitting

specifically if the network is too complex compared to the data.

Classification Algorithms

Classification algorithms are a family of supervised learning tasks in which the models try to predict the label of the given input data. In Classification algorithms, the output is a categorical variable as opposed to the numerical variable of regression. Most well known classification algorithms include Logistic Regression, Support Vector Machines (SVM), Decision Trees, Neural Networks and Random Forest. For the models explained in Regression section, the difference between the classification version and the regression version of these models are the type of output variable.

2.2.2 Imitation Learning

Imitation learning is a field of Machine Learning in which the model is trained to replicate expert behavior and parallels were drawn between IL and human learning by Barbierato et al. [25]. The advantage of IL is its ability to learn from demonstration only without requiring a objective function or reward/loss functions. The most straightforward IL approach is called "Behavioral Cloning" which uses a Supervised Learning model with expert demonstration data to learn and mimic the behaviour of the experts or systems by learning from the state-action pairs. In Foster et al., Behavioral Cloning method was analysed mathematically and concluded to be fundamentally harder than general Supervised Learning problems [26]. Imitation Learning is used as an alternative to Reinforcement Learning for some cases in which the objective function may be unknown or reward function is complex to model.

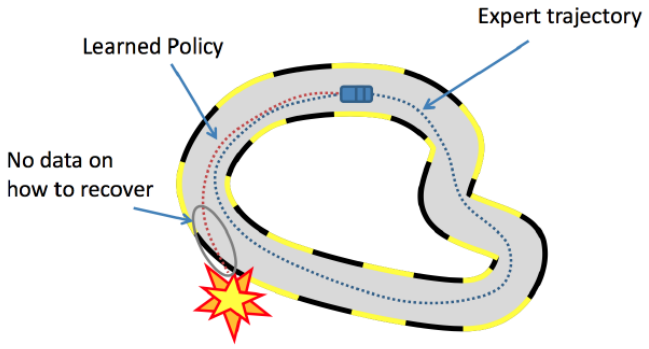

Figure 2.9: Imitation Learning for autonomous driving scenario.

Imitation Learning is used extensively in autonomous driving and driver modelling problems, and it is at times favored over optimization or RL algorithms for its simplicity. However a downside of IL is the lack of data for new and unseen scenarios. In the context of autonomous driving, the IL model will only be able to replicate the best behaviour from the expert demonstration but will not have any information about unseen situations such as going off-track or encountering a vehicle problem as noted by Lorincz et al. in Figure 2.9 [27].

Challenges of IL

- Generalization: The trained agent may fail to perform in unseen scenarios.
- Compounding Errors: Small errors or largely repeated behaviours in the demonstration can accumulate or directly influence the agent and cause a loss of performance, this phenomena is called "Bad Feedback Loop".

Advantages of IL

- Efficiency: IL is faster than RL or optimization for most use-cases because it learns from demonstrations and not by trial and error process.
- Simplicity: The working principle of IL is straight-forward and understandable.

To remedy the challenges faced when applying Behavioral Cloning method, two novel approaches are developed. "DAgger" [28] and "Generative Adversial Imitation Learning" [29]. DAgger Algorithm is very similar to BC, however in this method on-policy demonstrations are collected as the IL agent learns and the dataset of demonstrations is aggregated. In GAIL, there is a discriminator function that aims to distinguish expert actions from the learned policy whilst simultaneously learning the policy.

2.2.3 Optimization

Optimization is finding the optimal set of parameters for a function to minimize (or maximize) an objective function. Optimization is an important tool for decision making and in analysis of systems for it allows to find the best course of action to optimize a system or a problem. Main components of an optimization problem are the objective function, independent and dependent variables and mathematical constraints for the variables. The objective function depends on the characteristics of the problem defined by the variables. There are various types of optimization

algorithms depending on the nature of the problem or the purpose. The algorithm used in this thesis is Gaussian Process Bayesian Optimization.

Gaussian Process Bayesian Optimization is an intricate and complex probabilistic global optimization method that focuses on capturing the randomness and complex non-linear relationships. This method performs best with noisy and non-linear datasets because of the adaptation of "Gaussian Process" as the prior distribution. In GPBO, concepts of Bayesian Optimization is combined with Gaussian Processes and the resulting model is capable of dealing with black-box problems [30]. In black-box models, the explicit notation of the objective function is either missing or partially given. GPBO was selected for the optimization task presented in this research to overcome the lack of explicit objective function for performance outputs and driver input features.

2.2.4 Statistical Tools and Metrics

Standardization is used before training and testing a model. This processes makes sure the features are re-scaled so that their distributions fit between 0 and 1 however the inherent distribution isn't changed. Standardization makes sure the scales of each feature is the same given that different scales can disrupt the performance of Machine Learning models.

Formula for standardization is:

$$z = \frac{x - \mu}{\sigma}$$

Where: x is the data value, μ is the mean of the feature, σ is the standard deviation of the feature.

Feature Selection Methods

ANOVA (Analysis of Variance) statistical test is commonly deployed before inference to compare the means of different groups to make informed decisions about feature importance [31]. The outcome of ANOVA test determines each feature's importance on the output variable by quantifying the variance between the means of two or more groups (groups presented in the categorical output variable).

The F-statistic, which is the ratio of the variances of different groups, is used to determine if the differences between groups are statistically significant. This method is frequently used for conducting feature selection [32].

The F-statistic is calculated as:

$$F = \frac{\text{Variance of Group 1}}{\text{Variance of Group 2}}$$

For ANOVA, the F-statistic is:

$$F = \frac{MSB}{MSW}$$

Where: - $MSB = \frac{SSB}{k-1}$ (Mean square between groups), - $MSW = \frac{SSW}{n-k}$ (Mean square within groups).

Another method for feature selection is Pearson's Correlation Coefficient, r, measures the linear relationship between two continuous variables and is calculated as [33]:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Where: - x_i and y_i are individual data points, - $\bar{x}$ and $\bar{y}$ are the means of x and y, respectively.

Pearson's r ranges from -1 to 1, where: - $r = 1$ means a perfect positive linear correlation, - $r = -1$ means a perfect negative linear correlation, - $r = 0$ means no correlation.

Generalized Extreme Studentized Deviate (GESD) is used for detecting outliers in a dataset which is an important data preprocessing step since outliers can lower the performance of many ML models [34]. The test calculates the test statistic R_i for each observation, and compares it to a critical value from the Student's t-distribution. This comparison is done iteratively for each observation until there are no more data points left that exceed the threshold. GESD performs better than some other Outlier Detection methods in real-world datasets because the number of outliers are not fixed and is dependent on dataset-specific statistics.

Cross-Validation is a statistical tool to estimate the performance of models. The resulting performance usually has lower bias than other methods. The parameter k is the number of splits that the dataset will be split into. The most common 5-fold cross validation works by splitting the training set into 5 folds and performing validation on these folds. Afterwards, the final testing is done on the test set.

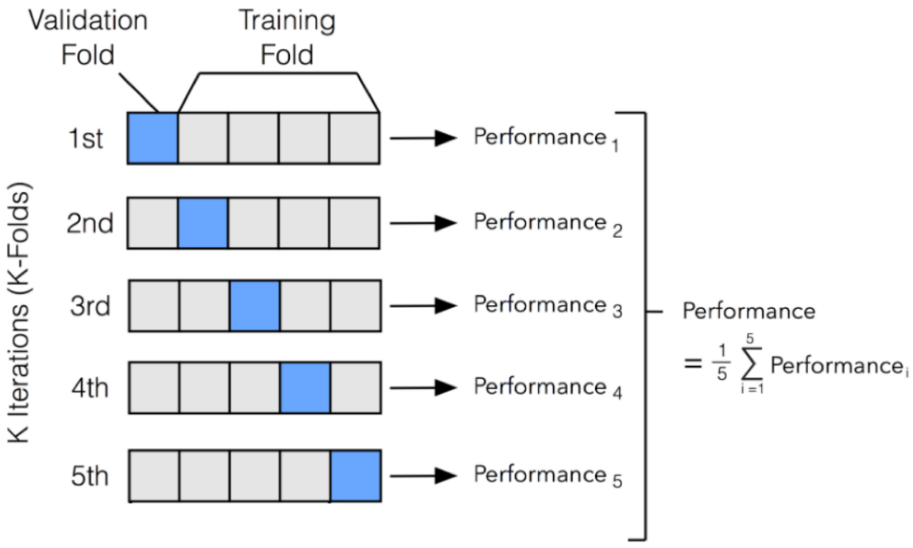

Figure 2.10: Visual explanation of 5-fold cross-validation

Metrics for Performance Evaluation

Regression Metrics	**Classification Metrics**
MSE (Mean Squared Error) $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$
RMSE (Root Mean Squared Error) $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	**Precision** $\frac{TP}{TP+FP}$
MAE (Mean Absolute Error) $\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	**Recall** $\frac{TP}{TP+FN}$
R-Squared (R^2) $1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$	**F1-Score** $2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

Table 2.2: Regression and Classification Metrics with Formulas

2.3 Machine Learning for Motorsports

2.3.1 Outcome Prediction

One of the most well-researched topics in the field of motorsports is outcome prediction. This focus is further fueled by the increasing monetary investment in formula racing and the previously mentioned need for technological advancement to make sense of the vast amount of data being collected. Outcome prediction for motorsports includes analyzing vehicle performance, driver behaviour and environmental factors. This section reviews the current methods and technologies used in outcome prediction, identifies gaps in the research, and discusses their relevance to the work presented in this thesis.

Research in outcome prediction typically involves ML models, regression algorithms, to predict race results based on historical data and for risk aversion [35]. Regression models are commonly used because they allow for continuous output prediction and are simple yet powerful tools Algorithms such as Linear Regression, LASSO, RIDGE, SVR, and Decision Tree Regression have been employed in Tulabandhula et al. [36] to predict the loss of rank for each lap. ANN's are frequently used in formulating outcome prediction problems as "Win, Loss" classification models. In the research of Thabtah et al. [37], the prediction problem is formulated as a two-class classification and they also proposed a feature selection step that combines two sets of features. One is an expert-selected subset of features, decided upon by domain knowledge. The second is a mathematically derived feature subset that uses feature selection methods. By combining the two subsets, it is ensured that final input features are both logical and suitable to the context. This approach was also adapted in the proposed thesis, to make sure there was no loss of contextual information. In the work of Stoeppels, ANN's are used to predict race results of F1 races from 2016 to 2017 for 4 drivers. As a result it is found that ANN model performs similarly to Logistic Regression for the specific problem [38].

SVM has also been widely used in motorsports outcome prediction. In Tejada et al. [39] SVM was applied to find the optimal pit stop strategy during F1 races. By framing the problem as a classification task, the model predicted whether or not a lap was the best time to change tires, based on factors like car performance, track conditions, and driver behavior. Interestingly, the study found that SVM performed better with the original dataset compared to one reduced via Principal Component Analysis (PCA), indicating that PCA's dimensionality reduction might have discarded important features as it was also discussed in [37].

Studies like Sicoie et al.[40] implemented Random Forest, Gradient Booster Regressor, and SVR models to predict championship standings and race winners. The feature selection step played a key role for this research, with SVR performing

the best in this study. Random Forest, another popular algorithm, has been used in Kumar et al. [41], where it outperformed other models like Logistic Regression and Decision Trees in predicting driver and team performance.

Research in NASCAR Racing domain such as [42] [43] by Pfitzner et al. investigate the impact of features on the output variable, which is the race outcome. They apply correlation analysis on the input feature set, and then perform an importance analysis to understand which are the most critical subset of factors for race performance.

Feature selection, particularly in complex motorsport datasets, has been a critical point in all of the research in this domain. Many studies, [42] [36] [40] underline the importance of accurate feature selection to improve model performance. The balance of removing unrelated or correlated features and keeping a certain level of variance and information is essential for predicting race outcomes effectively, as removing too many or contextually important features can degrade model performance.

There exist gaps in the research for ML models for sports outcome prediction, and the most evident one being the lack of real-time applications. Although with using historical data, valuable insights are uncovered and accurate predictions are made, motorsports inherently requires on-site, fast-paced applications.

Existing research predominantly focuses on Formula 1 or NASCAR races, leaving lower-level competitions unexplored. The reason for this natural preference for high-level motorsports competitions is presumably because there is easier access to data and there is more funding. Also, FS competitions are not suitable for pit-stop decision making problems or multi-driver race winner prediction. However, the basic purpose of predicting output variables such as corner time, lap time or vehicle speed are research directions common also for the FS domain.

2.3.2 Driver Modelling and Strategy Optimization

Driver modeling and optimization of race strategies is critical for success. For many years this has been done by human intellect only, engineers with historical data in their minds applied the seasoned race strategies they formulated to races. With the development of data-driven approaches, now engineers and software work together for race strategy optimization. This task is approached in two main ways; driver modeling or strategy optimization as an outside entity such as a virtual race engineer.

The research and technologies developed for driver modeling and strategy optimization revolve around three main topics; Numerical Optimization algorithms,

Reinforcement Learning and Imitation Learning.

Driver Modeling

Driver modeling is the process of creating a smart agent, usually modelled by ML methods, that learns a policy from past data and demonstrations and can imitate or make decisions by itself in certain conditions. It can also be any research in which a driver and his/her behavior is being quantified and classified to be then tested in new settings.

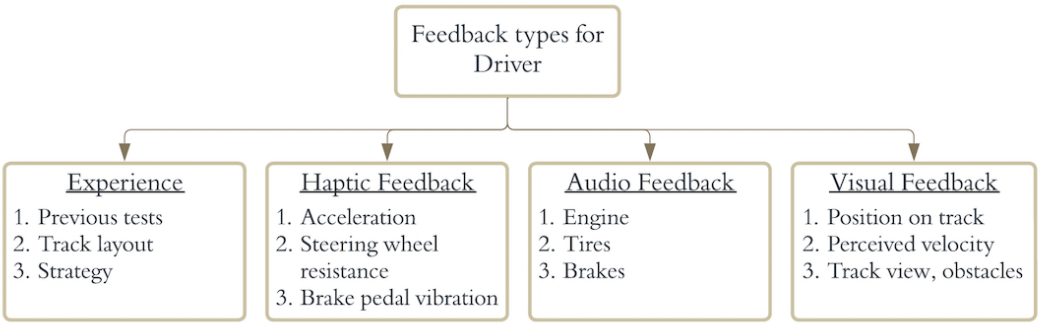

Figure 2.11: Factors influencing driver behaviour.

Driver modeling plays a pivotal role in motorsports, where understanding and imitating driver behavior can lead to significant performance improvements as it increases the efficiency of testing procedures. The ability to model individual driving styles and their effects on vehicle dynamics allows teams to tailor vehicle setups and strategies and test them with these driver models. This section reviews the technologies and methods used in driver modeling, highlights the gaps in current research, and discusses their relevance to the thesis.

Having a deep understanding of the style of the driver is critical in motorsports, regardless of the level of competition. If the team can quantify or underline different styles and behaviors of their drivers, then they can more easily give feedback or understand what to optimize. In von Schleinitz et al. [44], it was found that top-level drivers can be distinguished with high accuracy by analysing only the brake and throttle pedal signals from data of a single corner. Following from this result, it can be said that from looking at driver control inputs, it is possible to see a distinct style.

An approach to tackle these challenges is presented in the work of Segers' "Analysis Techniques for Racecar Data Acquisition Second Edition". [45] Aside from being a fountain of information and invaluable for the proposed thesis, it is

also a guidebook on how to extract indicators of driver performance in order to analyse the style.

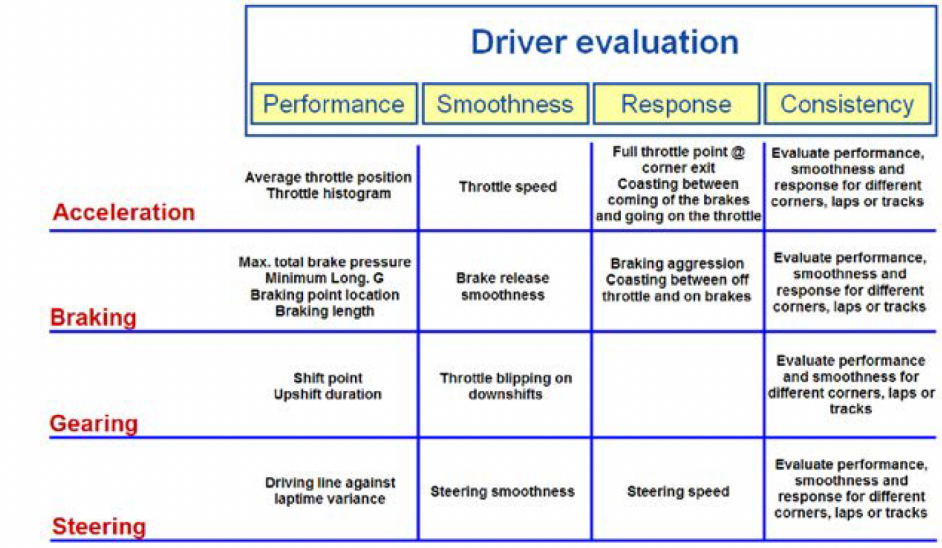

Figure 2.12: Driver evaluation indicators.

In the aforementioned book, Segers divides driver performance indicators into four main titles; Performance, Smoothness, Response and Consistency. Every indicator given in Figure 2.12 has been adapted to the specific context of this thesis work and used extensively in Chapter 5: Driving Style Analysis except for the ones listed under "Gear" given that SCP vehicle is an electric vehicle and doesn't contain gears.

In Hojaji et al. [46], drivers are segmented into three classes depending on their lap times, and various remarks are made about the style of each segment. After conducting a detailed driving style analysis step, it is seen that prominent styles exist for most of the driving behaviours, and further research can be made for analysing braking and accelerating points, corner segments and discovering the connections between all metrics that define the parameters to describe driving style. SVR, XGBoost and Random Forest Regressor were implemented to select features and predict lap times.

Simply wanting to see the style and believing there is one is not enough to concretely prove that there is a style. It is a job easier said than done, mainly because of the following reasons summarized in Lockel et al. [47]:

- Resulting behaviour is influenced by random outside factors
- Driver behaviour in itself is random due to self-adaptation
- Similarly performing drivers can have highly distinct styles
- Minor differences in style can result in having major impacts on outcome, deviating the scale of variability

The second factor, which is the randomness of human behaviour, is the most critical point. Drivers may have a style, but from one lap to another they can adapt to the vehicle settings, therefore they may act differently. Other than adapting, a driver might be actively learning, hence bettering his/her style. These add into the inherent stochastic nature of human beings, for this reason a suitable driver model needs to be robust, should mimic the complexity of the human driver and yet be explainable and reproducible. [47]

One notable study by Braghin et al. [48] proposed a two-step approach; a dynamic optimization challenge to identify an optimal trajectory and a quadratic minimization problem to optimize steering, braking, and throttle behaviours. An important take-away from this research is the trade-off between taking a corner with high speed yet longer path and low-speed with a shorter path. If the high speed approach is taken, the driver must take the highest curvature radii to maintain speed. To balance this, the optimal trajectory solution finds a compromise between the two, and shows how the driver behaviour is heavily influenced by the vehicle dynamics.

Various studies implemented Neural Networks for imitating driver behaviour. [47] [49] This architecture is called Imitation Learning, and by implementing NNs to gather information and patterns from expert demonstration to form decisions by imitation is specifically Behavioural Cloning.

Another way of approaching the question at hand is by using Optimization algorithms such as Genetic Algorithm, Particle Swarm Optimization and Ant Colony Optimization. These methods were explored in the work of Benderius [50], to optimize driver control inputs.

Strategy Optimization

Reinforcement Learning, Imitation Learning, Optimal Control Approach, Dynamic Programming, Markov Decision Process and Numerical Optimization methods are the main fields of research for race strategy optimization. In some cases these methods are used in combination or comparison, to enhance the efficiency or performance of the overall model.

Reinforcement Learning is used in various researches due to its adaptability to any situation in which an agent must learn through trial and error, and minimize the penalty. RL enables AI to iteratively move in the action space, mimicking a driver's adaptation to his/her environment and vehicle. Deep Q Learning method was used in GT Racing context in Boettinger et al. [51] where it is used to refine strategy decisions. In other research papers, RL enabled the researchers to create models that achieve better performances than human drivers, and adapts to unseen tracks. [52] [53]

Imitation Learning is used extensively in research on race strategy optimization. In Heilmeier et al. [54], dual-ANN architecture is chosen to optimize race strategy by choosing the expert data-inspired action examples such as pit stop decision and tire compound choices. First ANN chooses whether to pit in the current lap, and if the decision is positive then the second ANN decides which tire compound to fit to the vehicle. The researchers conclude the "Virtual Strategy Engineer" gives reasonable and understandable decisions generally, and the fast-response is an advantage in real-time scenarios.

A research on the FS race strategy/performance optimization domain by Jimenez et al. [55], also highlight the lack of ML applications in this domain. These researchers are in collaboration with California State Polytechnic University Pomona, student-lead Formula SAE team Bronco Motorsports to remedy the "Data Underutilization" problem they face. First research idea was applying Linear Regression to predict throttle pedal position, steering wheel angle and brake pedal position given time and position. However they have realized as it was noted also in this thesis work that the relationship between these variables are highly non-linear and there needs to be a more suitable model to grasp the full scale. By utilizing ANNs, they achieved 70% accuracy with using time and positional information in predicting the driver control behaviour. This research work is also an example of Imitation Learning, since the expected output is the inputs the driver puts into the vehicle.

There are research work combining ML methods with optimization and heuristics methods to capture the full variability of the problems at hand. In Pontin et al. [56], an AI-based race strategy assistant is developed by first deploying a NN to optimize the pit-stop decision, followed by a Mixed-Integer Linear Programming (MILP) model to select optimal tire compound to fit. Another approach is discussed in Liu et al. [57], this research leveraged Deep RL and Markov Decision Processes to develop strategy optimization framework. The focus is on managing energy and thermal effects for Formula-E races. Comparing this approach with MCTS, the researchers found DRL and MDP performing superior and underlined the importance of modeling the action space as a continuous environment. Non-Linear Model Predictive Control (NMPC), paired with Artificial Neural Networks (ANNs), was used to optimize real-time race decisions like braking and acceleration. [58]

Monte Carlo Methods were applied to optimize pit stop timing and tire choices by simulating probabilistic driver performance. These researches leveraged Markov Decision Processes (MDPs) and Monte Carlo Tree Search (MCTS) for modelling sequential decision-making for strategy optimization problems, incorporating Dynamic Programming and Temporal Difference Learning for more complex scenarios. Together, these approaches enhance race strategy by accounting for uncertainty and improving decision-making in critical race moments. [59] [60]

Chapter 3

Data Collection and Analysis

This chapter outlines the methodology followed for data collection, preprocessing, and analysis. It is a known fact that data scientists spend a considerable portion of their project time on preprocessing the data. This is very real and also needed, since the integrity of the data plays a pivotal role in ensuring the quality of subsequent modeling and decision-making steps. By following the systematic approach explained in Figure 3.1, the aim of this step was to extract valuable insights and accurate data to build the following steps on.

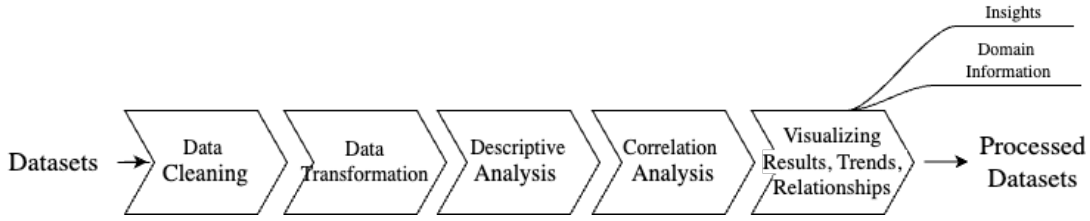

Figure 3.1: Data Preprocessing pipeline.

Throughout this year, the team conducted countless simulator tests to gather information about the new vehicle they have designed. The driving simulator provides a highly efficient and cost-effective method to gather performance data before conducting physical tests on the vehicle.

The data collected during these simulations include driver inputs—such as steering, throttle, and braking, as well as the vehicle's dynamic responses, including speed, acceleration, and vehicle dynamics metrics.

MATLAB is used extensively to preprocess and visualize data, allowing detailed analysis of the vehicle's behavior and helping identify optimal configurations. This section details the data analysis process and the preparation and organization of the datasets.

3.1 Simulator Environment

SCP uses "VI-Grade Car Real-Time Simulation" driving simulator that allows the creation of a digital twin of the real vehicle and the local test track Cerrina is also constructed in this environment to test the car setup configurations before actual testing of the car on track. Creating this virtual environment where SCP engineers can test and validate vehicle designs and configurations before physical prototypes are built is crucial for the team since the team has to create a new vehicle each year, therefore efficiency is of the most important value.

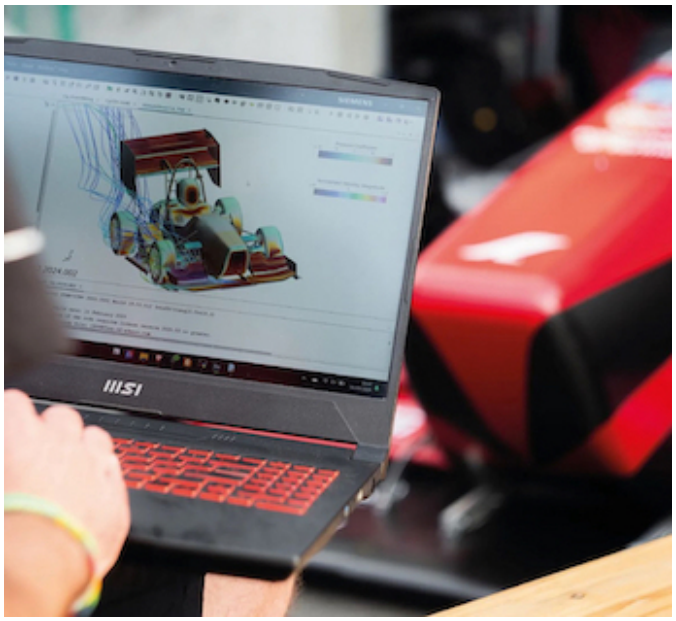

Figure 3.2: Driving simulator ouput of SCP.

VI-GraphSim, VI-DriveSim and VI-CarRealTime products are used in combination to create the most comprehensive Driver-in-the-Loop (DIL) simulation technology. These tools allow SC drivers to interact with the simulation environment in real time, providing feedback on the handling and vehicle response. MATLAB-Simulink connection provided by the V-model is the key for SCP, this link allows the vehicle model to exist in Simulink environment.

As hardware, Tower WorkStation Dell Precision 7920 XCTO, DELL Monitor 24" G2422HS, Fanatec Clubsport Universal Hub V2 Steering Wheel, Fanatec Podium Wheel Base DD1 and Fanatec Clubsport Pedals V3 are used by SCP.

In this thesis, the collected data from VI-Grade Car Real-Time Simulation simulator is employed for all of the following activities. The data collection is dependent on the sensors attached to the simulator, which collects data from the

driver input controls, such as the steering wheel, brake and throttle pedals, and the response of the vehicle, such as the resulting speed, longitudinal and lateral acceleration and so on. The sampling frequency is 100Hz, which corresponds to 100 samples per second. This is a high sampling frequency that ensures data accuracy and usability.

Squadra Corse PoliTo Vehicle Dynamics division has been conducting data analysis for their activities and have an existing level of detailed collection and transformation workflow for data processing. Collected data has the same level of complexity as real track data if not higher, and the V-Model allows tuning parameters of the vehicle model according to the results of the simulator stints.

Figure 3.3: Cerrina track layout.

There are 10 corners in the Cerrina track however the last 3 are more accurately classified as chicanes. Chicanes are slightly curved lines on a track and they require separate attention whilst conducting a corner-based analysis. In chicanes, the driver almost never brakes fully, and uses throttle to accelerate, hence the behaviour is more similar to a straight line rather than a corner.

3.2 Datasets

There are two datasets created and supplied for this thesis by SCP Vehicle Dynamics Team; Simulator Signals and Corner Indicators Dataset. Two datasets were updated regularly with data from new drivers and stints, depending on the simulation tests

ongoing during the 23/24 season.

Simulator Signals Dataset is a direct product of the driving simulator and the data stream belongs to the stints of SCP drivers. There are 8 drivers in total represented in this thesis, their names are redacted for anonymity purposes and called Driver A, B, C, D, E, F, G and H. The Vehicle Dynamics division collects and stores the data, and preprocesses it to then use as input for their Data Transformation pipeline. Using MATLAB, the team produces the second dataset as filtered indicators extracted from the first dataset. Most of the indicators at this step has been inspired and realized by consulting "Analysis Techniques for Racecar Data Acquisition Second Edition" by Segers [45].

3.2.1 Simulator Signals Dataset

Simulator Signals Dataset is a time-series dataset with over 1 million data points, containing driver control input variables, vehicle dynamics state variables and output variables. In this dataset, there are signals directly coming from the simulator, and variables created from the signals such as "LapTime".

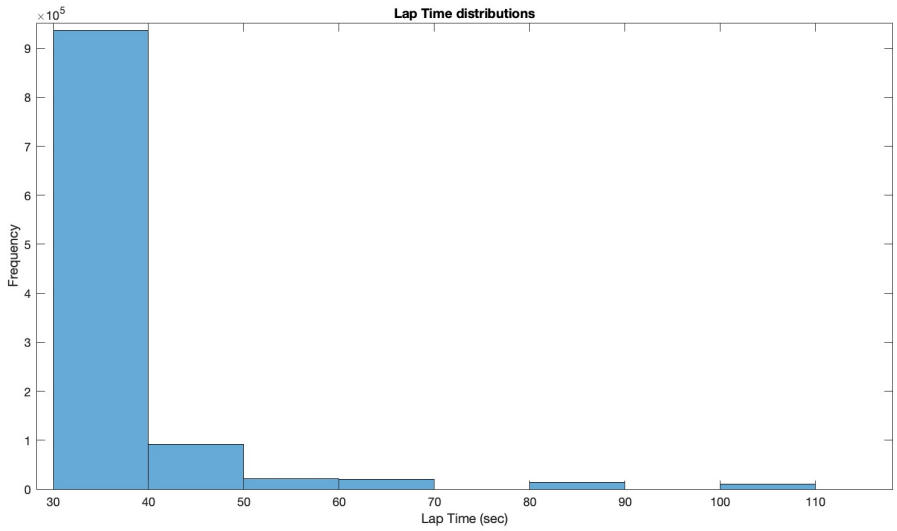

Figure 3.4: Lap times histogram.

Dataset Variables Summary

- "Brk_Input" is the percentage of movement of the brake pedal. If the driver is applying full pressure on brakes, this value will correspond to -100.
- "AccX": is the longitudinal acceleration (m/s^2) and refers to acceleration in a straight line.
- "AccY" is the lateral acceleration (m/s^2) and it measures cornering performance.
- "Throttle" is measured the same way as brake input as the percentage of movement from the throttle pedal.
- "SteeringAngle" is the degree of which the steering angle is turned from its resting balanced position. This value has values in the (-100,100) range, and positive value means a turn to the right.
- "LapTime" is calculated a the time it takes for the driver to cross the finish line, and it is measured in seconds.
- "V_CG_1" denotes vehicle speed at each instance (m/s).
- "PosX and PosY" variables are positional values that locate the vehicle on the track by giving its coordinates.
- "RPM" values denote each of the wheels rotations per minute.

1. **Data Cleaning**
    - *Identifying and Handling Missing Values*: In this dataset there were no missing values since it is an interrupted stream of time series data.
    - *Outlier Detection and Treatment*: Outliers were present, however they were not removed because of two reasons; the outliers were not extremely different then original data points and they carry information about rare actions the drivers took.
    - *Ensuring Data Integrity*: All crucial variables were compared against multiple laps to ensure there were no simulator signal system malfunction. For driver error, there were variables used as flags in scenarios in which the driver went over track limits, or touched a pole. These actions mean that lap is invalid and unusable, therefore the data points with positive error flags were filtered and removed. The remaining cleaned dataset is accurate and only contains valid timed laps, providing a foundation for reliable analysis.

2. **Data Transformation**

- *Scaling and Normalization*: Scaling was done before each ML model however in the preliminary phases the datasets aren't scaled to preserve the data integrity and explainability to ensure contextual accuracy.
- *Feature Engineering*: The categorical feature "Date_Driver_Stint#" was split into its components, Driver name was extracted to be used as a new and separate categorical feature named "Driver".

Correlation Analysis

In order to make sure unique information is presented in each of the variables, correlation analysis was conducted on Simulator Signals Dataset.

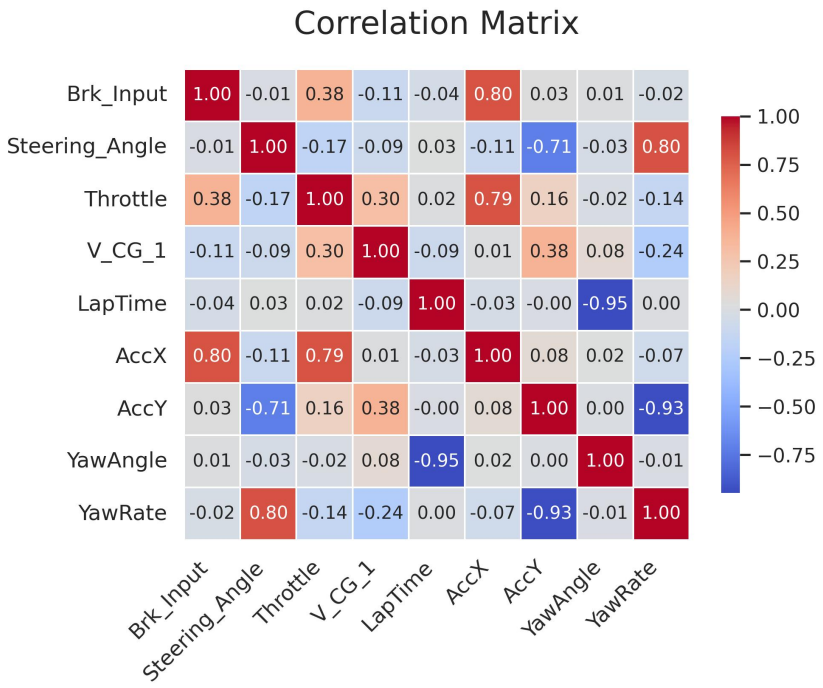

Figure 3.5: Correlation Matrix for subset of features.

To achieve this, Pearson's Correlation Coefficient method was implemented. The full results of the analysis is given in Appendices A. The strong positive correlation between "Brk_Input" and "Throttle" with "AccX", or the strong negative correlation between "Steering_Angle" and "AccY" is completely explainable by the nature of these variables. Braking and acceleration actions are related to

the longitudinal forces on the vehicle, and inversely steering is related to the lateral force. "Steering_Angle" is positively correlated with "YawRate" and this relationship is also a phenomena coming from vehicle dynamics. Yaw refers to the rotational movement of the vehicle, and steering action is creating the yaw effect, hence the positive relationship. "YawAngle" has a strong negative correlation with "LapTime". As a conclusion of this analysis, "YawAngle" and "YawRate" will not be considered in the Outcome Prediction framework given in Chapter 4.

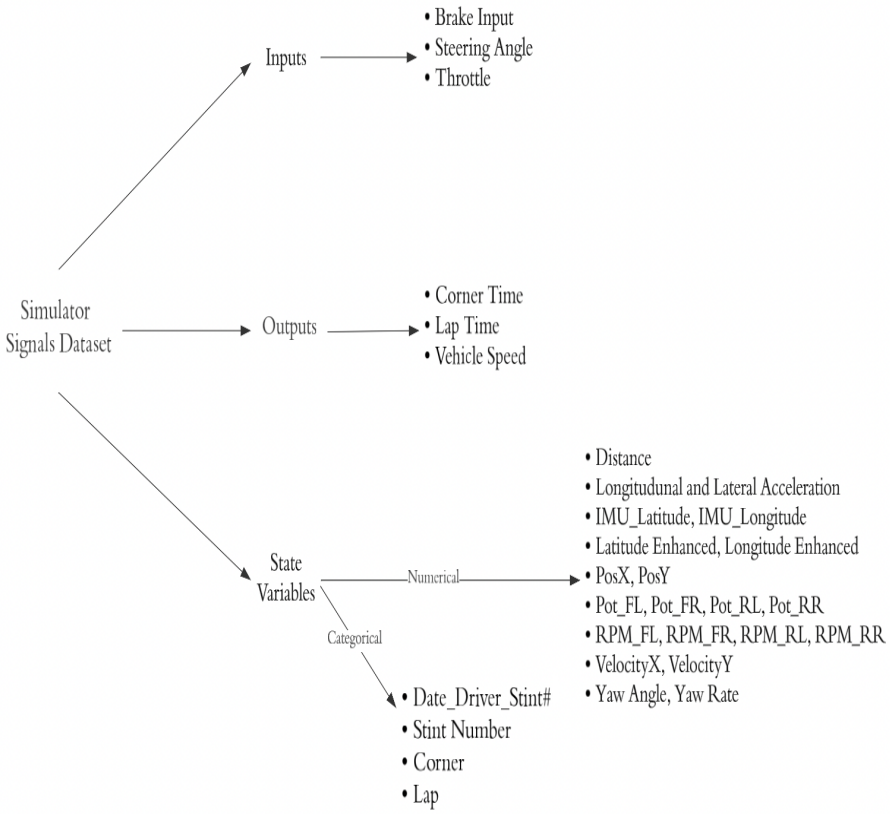

Figure 3.6: Simulator Signals Dataset final structure.

3.2.2 Corner Indicators Dataset

Corner Indicators are extracted by the Vehicle Dynamics team to offer insights of driver behaviour and vehicle performance per corner. Simulator Signals Dataset is filtered for the 7 corners following the data processing framework created on MATLAB by the team, and important information is extracted. Chicanes are not presented in this dataset, namely Corners 8 9 and 10, since they require separate analysis that doesn't coincide with characteristics of corners.

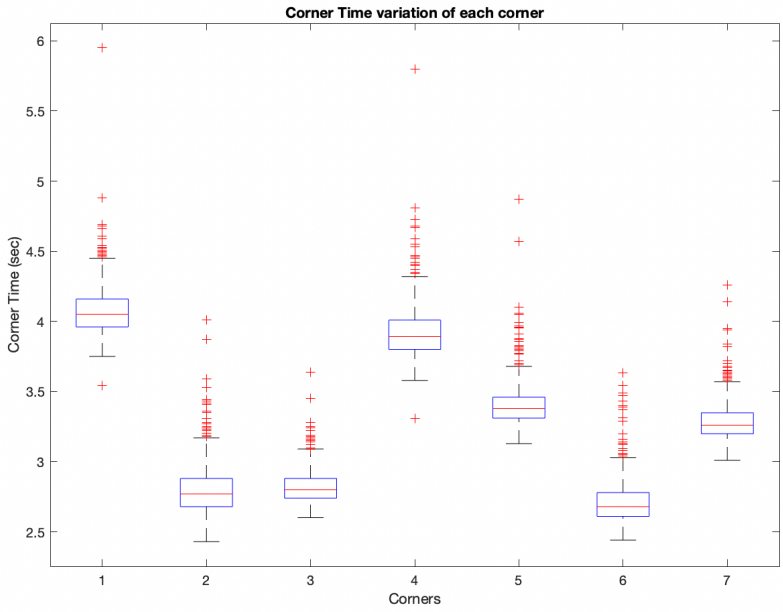

Figure 3.7: Corner Time variation of each corner.

Figure 3.7 provides insights about driver performance across different corners. It can be seen that the times vary significantly, and this supports the visual analysis of the track shape. Corners 1, 4, 5 and 7 have higher median corner time, suggesting longer or lower speed corners. Corners 2, 3 and 6 have lower median corner time and lower variability, suggesting that they are easier to handle. The presence of high variability and outliers for Corners 4, 5 and 6 may suggest the possibility of drivers struggling with these corners, resulting in inconsistent times. In the following list, definitions and calculations of the variables will be given with courtesy of the Vehicle Dynamics Division of SCP.

Output Variables

- "Corner Time" is the time, in seconds, taken to complete a corner
- "Minimum Speed" represents the lowest speed reached in a corner m/s
- "Exit Speed" is the speed ("V_CG_1") in the exit of the corner.
- "Steady State Cornering Time" is the time spent whilst the vehicle is turning with an almost constant curvature radius.
- "Overall Understeer" is the average understeer angle of the tyres, filtered to be higher than a given value. The understeer angle is defined as $\alpha_u = |\delta_f| - L \times \left|\frac{a_y}{V^2}\right|$ δ_f [rad]: Average steering angle, L [m]: wheelbase, a_y [m $\cdot s^{-2}$] : lateral acceleration ,V [m $\cdot s^{-1}$] : Vehicle speed
- "Overall Grip Factor" is the measure of grip used by the tyres calculated as log(Combined_Acc) = $\sqrt{\log(\text{AccX})^2 + \log(\text{AccY})^2}$, then by filtering the combined acceleration values by a predefined minimum value, overall grip factor is created.

Input Variables

- **Steering Group:**
  - "Peak Steering Angle" is the maximum angle of steering during a corner.
  - "Steering Aggression" is the maximum value of the derivative of the steering signal.
  - "Steering Erraticness" is the average of the absolute value of difference between the "SteeringAngle" signal and its moving average. It is an indication of the smoothness of the drivers input.
  - "Steering Speed" is the mean value of the absolute value of the differential of "SteeringAngle" signal. It is an indicator of how fast a driver is using the wheel.
  - "Steering Integral" is the cumulative steering actuation during a corner.
- **Brake Group:**
  - "Peak Brake" is the maximum brake pressure applied during a corner.
  - "Brake Aggression" The speed or intensity with which braking is applied. Brake aggression is the peak derivative of the brake signal in the corner.
  - "Brake Erraticness" is the average of the absolute value of difference between the "Brk_Input" signal and its moving average.

- "Brake Speed" is the mean value of the absolute value of the differential of "Brk_Input" signal. It is an indicator of how fast a driver is actuating the pedal.
- "Brake Integral" is the cumulative brake pedal actuation during a corner.

- **Throttle Group:**
  - "Number of Throttle Peaks" is the number of instances where maximum throttle was applied.
  - "Throttle Erraticness" is the average of the absolute value of difference between the "Throttle" signal and its moving average.
  - "Throttle Speed" is the mean value of the absolute value of the differential of "Throttle" signal.
  - "Throttle Integral" is the cumulative throttle pedal actuation during a corner.

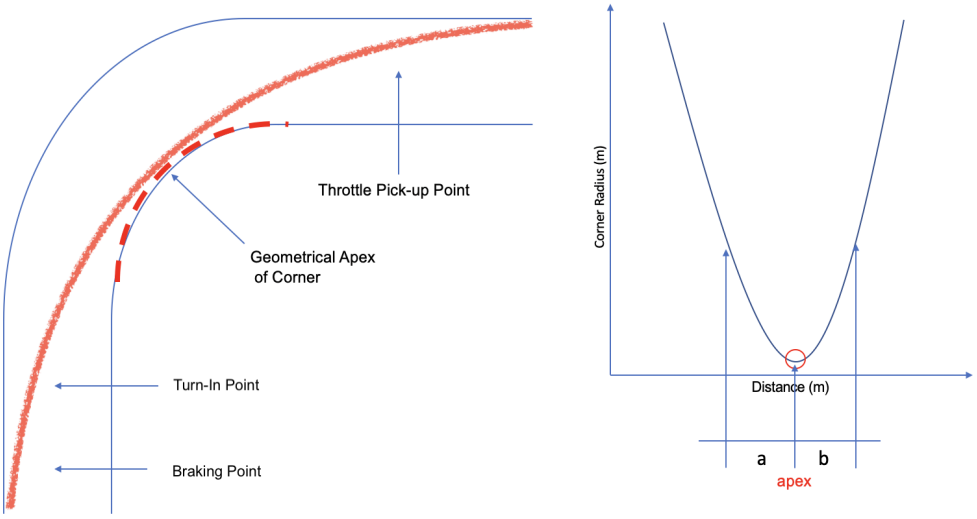

Figure 3.8: Cornering actions sequence and Apex Number.

- **Position Indicator Group:**
  - "Throttle Pick-Up Point" is the point of the corner in which the vehicle is turning while 3% of full throttle is being applied.
  - "Turn-In Point" is the point at which the driver begins to turn the vehicle into the corner at 20% of the maximum steering angle.

– "Braking Point" is the location where braking is initiated for a corner. It is measured by finding the time instances in which the "Brk_Input" equals its local maxima.

– "Brake Release Point" is the point where the brakes are released after the corner is taken.

– "Apex Number" % is calculated for driving line analysis. At the beginning of a corner the curvature radius is high, then approaching the apex it decreases. The minimum value corresponds to the apex, and after it increases again until the exit of the corner. In Figure 3.8, the distance from the corner entry to the apex is "a" and the distance from the apex until the exit is "b". Apex Number is calculated as $= a/(a+b) * 100$

– "Peak Curvature" (1/m) maximum curvature demonstrated by the driver in a corner. When cornering, a driver should aim to take the largest corner radius (smallest curvature) to reduce the corner time [45].

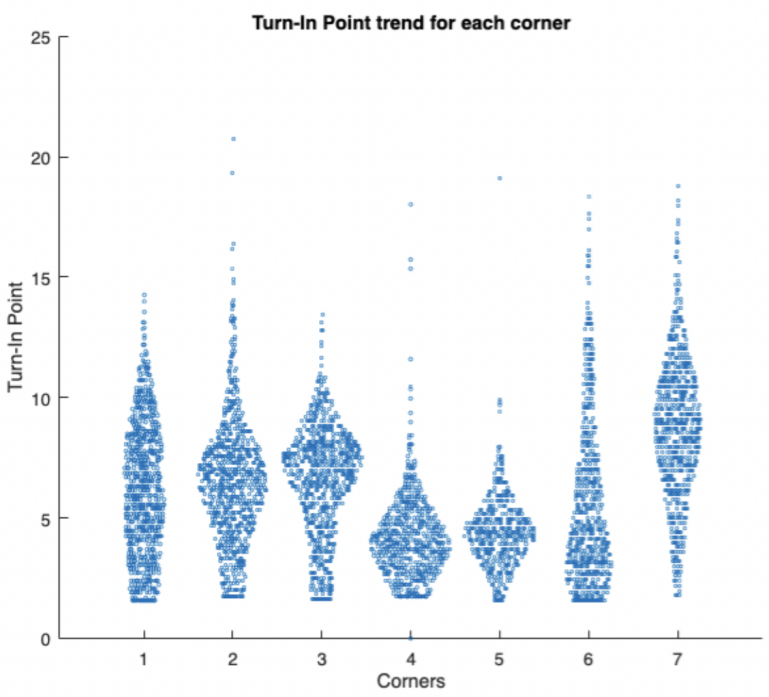

Figure 3.9: Turn-In Point trends for each corner.

1. **Data Cleaning**

   - *Identifying and Handling Missing Values*: MATLAB used to identify missing values. Brake group variables contained 15 missing values (NaN), and this occurred in parts of the track in which drivers didn't brake. After consulting with the team, it was decided to remove the rows containing missing values to preserve data integrity and contextual information. Also "Trail Braking Understeer & Grip Factor" variables had missing values because of their nature. Trail Braking means when after the driver brakes for a corner, he/she gradually releases the brake pedal, creating a trailing motion through the corner. These variables are created by filtering the Simulator Signals Dataset in a specific way in order to only take instances when the driver was using this technique. Therefore these variables had "NaN" for the rows in which the drivers weren't trail braking. These values were filled with Linear Interpolation.
   - *Outlier Detection and Treatment*: It was seen that methods such as "Moving Mean" or "Moving Median" fall short of capturing the patterns of a corner-based dataset. GESD (Generalized Extreme Studentized Deviate) was then used to detect outliers. "Peak Curvature" contained 20 outliers and "Steering Aggression" had 19 outliers. These outliers offer insight about corners and how drivers reacted on those corners, since they were results of and reactions to an extreme condition. The outliers were kept to contain as much information in the dataset as possible.

2. **Data Transformation**

   - *Scaling and Normalization*: As the previous dataset, scaling or normalization wasn't applied in this step, to preserve the original numerical scales.
   - *Feature Engineering*: "Stint Number" , "Corner" had character as type, they were transformed into categorical variables. "Date_Driver_Stint#" was split into different features to extract the "Driver Name" to be used for identifying the drivers.

Correlation Analysis

Correlation analysis was performed to investigate the relationships between key variables. Understanding the relationships between variables is crucial for predictive modeling, as it underlines and reveals which features have the strongest impact on overall performance. Analysis for correlation between output variables was also conducted to understand what subset should be chosen as output for the future models because the number of features in this dataset exceeds normal practice. Some of the variables have certain relationships inherently depending on Vehicle dynamics such as "OverallUndersteer" and "MinimumSpeed" or "ExitSpeed" have a moderately negative relationship. This is coherent with the vehicle's nature. If the driver and vehicle are experiencing more understeer, it is very likely that the vehicle lacks balance, and this will lower the speed.

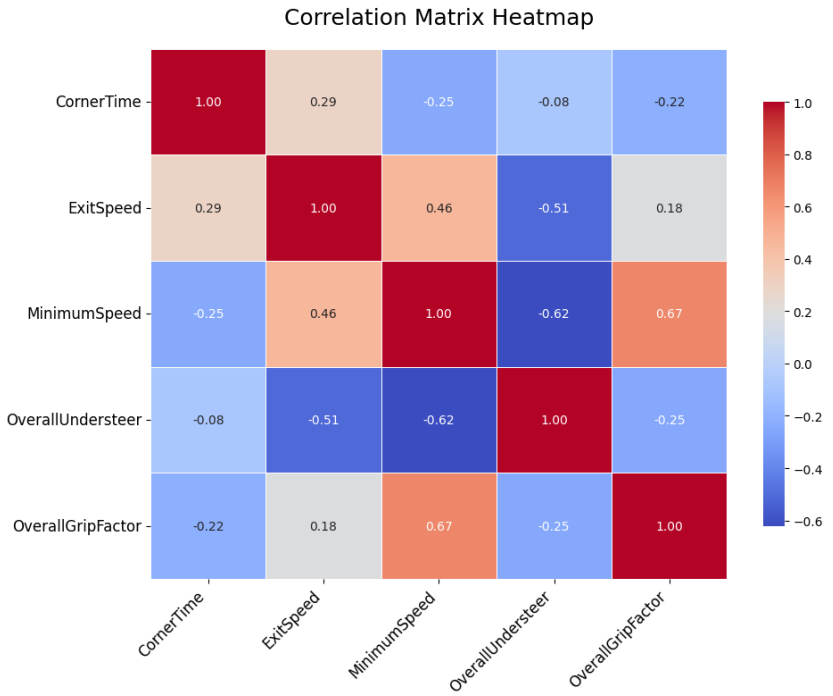

Figure 3.10: Correlation Matrix of output features.

There is no evidence of strong correlation between the output variables, all of the variables offer different information hence the output variables are kept. Many variables in Corner Indicators Dataset are calculated from the same signals by different aggregation steps. The main information is suspected to be present in multiple variables belonging to the same group and if a high level of correlation exists, the variables must be removed before being used as input for the ML models.

To remedy this, groups of variables are analyzed, and by adding domain knowledge, variables with high levels of correlation (>0.75) have been excluded from the dataset. The full results of this step can be seen in Appendices A.

Previous findings on the correlation of output variables are also visible in Figure 3.11, most relationships are visibly represented as dense clusters for each of the corners. For some of the output pairs, there exists also variance between corners and this can be caused by the difference in nature of corners. For example, "CornerTime" and "OverallGripFactor" features' data points form inversely related straight lines for most of the corners besides Corner 7, meaning the general performance for Corners 1, 2, 3, 4, 5 and 6 was heavily influenced by the grip factor in a negative way.

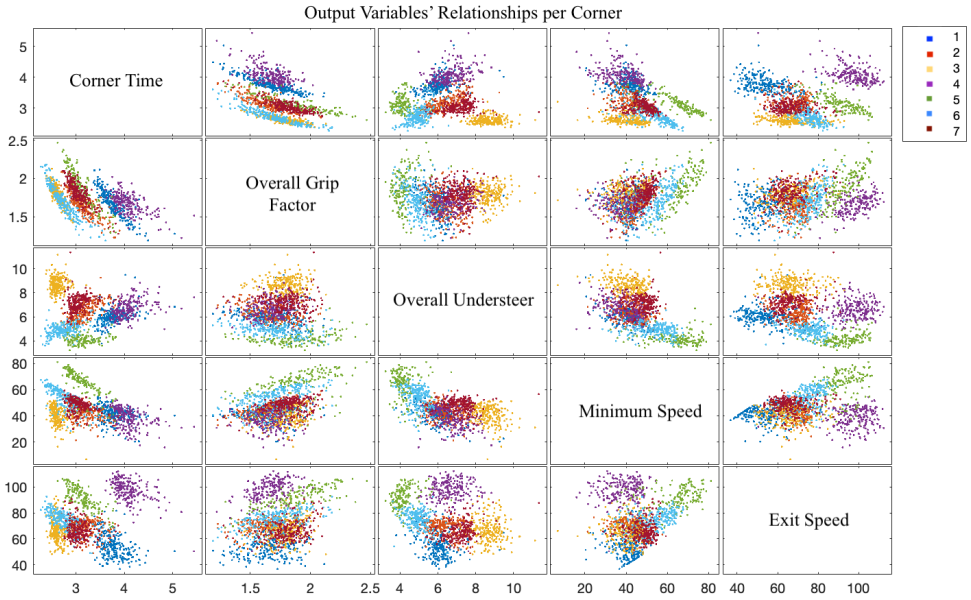

Figure 3.11: Relationships between output variables for each corner

In Figure 3.12, it can be seen that the highest "MinimumSpeed" values were recorded for all of the corners when there was more grip available for the tires and the "Understeer" values were low. This result is supported by the correlation analysis and knowledge from vehicle dynamics. The drivers were able to refrain from slowing down the car for cornering in situations where the car had enough grip to hold on to the track and didn't experience understeer.

Driving line and trajectory taken on a corner is defined by the curvature r,

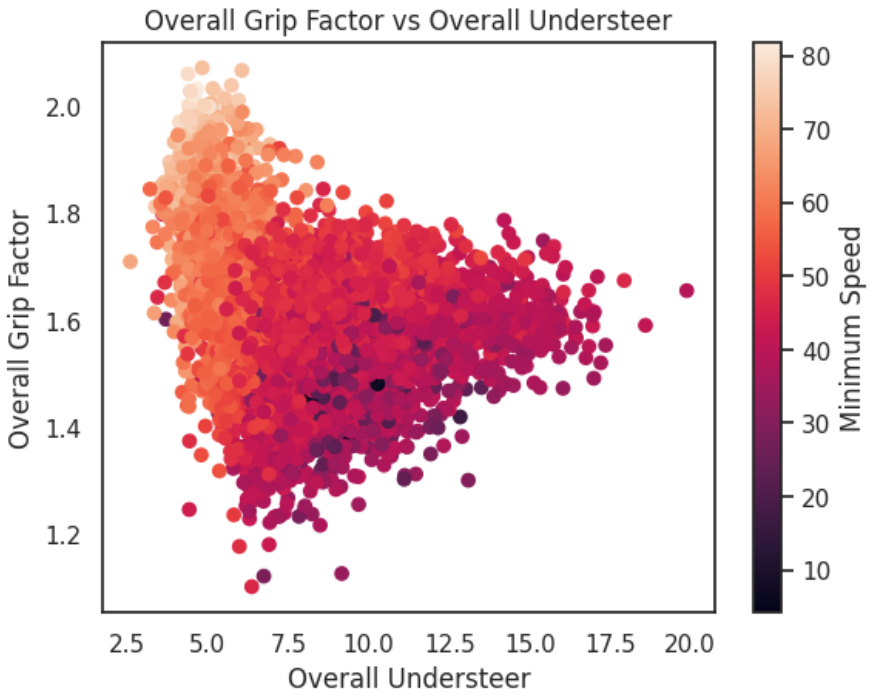

Figure 3.12: Understeer, Grip Factor and Minimum Speed Relationship.

which is the inverse of the corner radius R. Mathematically R is defined as:

$$R = \frac{V^2}{a_y} \tag{3.1}$$

Where:

- R = Corner radius m
- V = Vehicle speed m/s
- a_y = Lateral acceleration m/s^2

From Figure 3.13 and previous explanations of vehicle dynamics in Chapter 2, the relationship between curvature and corner time can be further analyzed as:

- An increased curvature signals the driver taking a tighter line through the corner. A higher curvature implies a higher deviation from a straight path, resulting in a higher steering angle and potentially lower speed in order to safely turn the car.

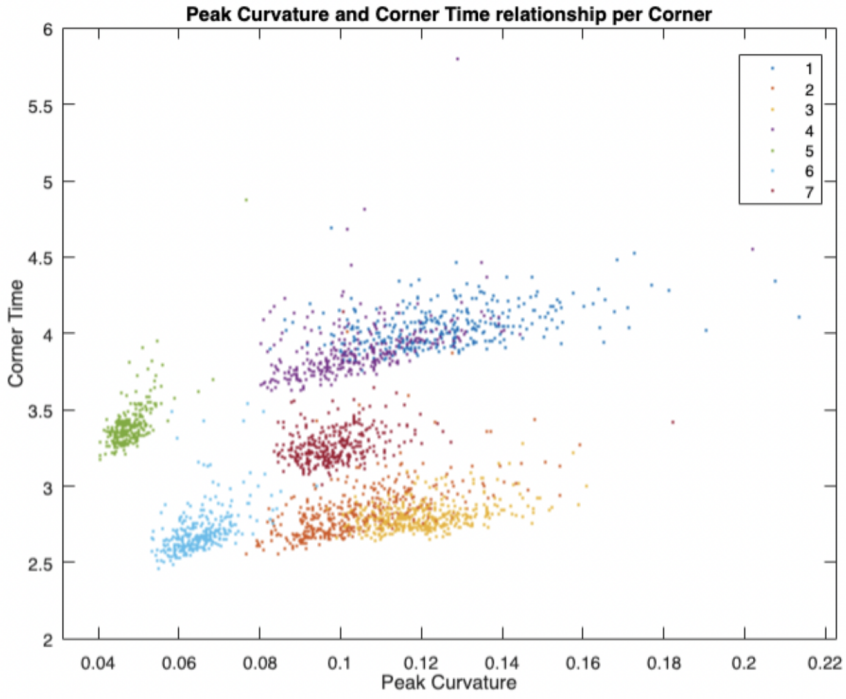

Figure 3.13: Peak Curvature and Corner Time values for each corner.

- If the speed isn't low enough, the driver may experience problems with the balance and control of the car, therefore taking a high curvature line is generally riskier compared to a lower curvature.
- Mathematically, if V is kept constant:

$$T_c \propto \frac{1}{R} \tag{3.2}$$

which implies that the corner time increases as the corner radius decreases and this claim is supported in Figure 3.13. For every corner, higher "PeakCurvature" values (lower corner radius) resulted in higher "CornerTime".

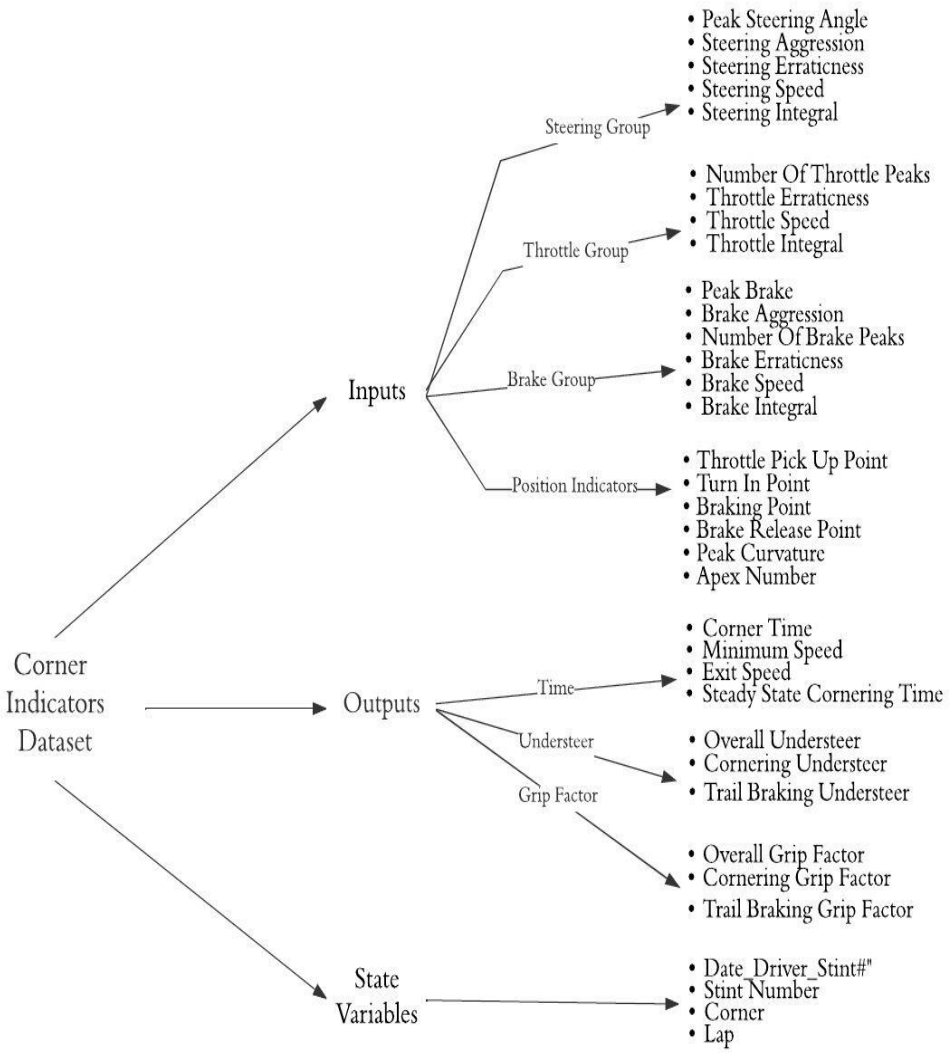

Figure 3.14: Corner Indicators Dataset final structure.

Chapter 4

Outcome Prediction

In this chapter, the results and findings from the regression models applied to both of the datasets will be discussed. Outcome prediction task was chosen as the first step of this research work for three reasons.

1. To analyse the datasets further and to achieve a more in-depth understanding of the underlying mathematical nature of the said data, regression models are used as a method to reveal this information

2. To supply Squadra Corse with regression models in order to help them predict and prepare for vehicle setups or test stints beforehand, granting them the possibility of foreseeing the errors or problems

3. To extract the regression line formula for the output variable Corner Time.

In Chapter 6, the extracted regression formula was adapted as the objective function for numerical optimization.

MATLAB Regression Learner Tool was chosen as the model building and testing environment for ease of integration of the datasets and repeatability, considering possibly in the future SCP will deploy the said models on MATLAB and Simulink.

Models built for this task are selected to encompass all families of Regression models and to have a stronger chance at finding the model best suited for these datasets. An early concern regarding this topic was the lack of information to supply the models on vehicle dynamics. Tire behaviour, aerodynamics and countless other physical factors play a huge role in the performance of the vehicle and this information wasn't present in the datasets. Also, it was seen that most of the variables had far-from-normal distributions, although there were trends the data-points rarely expressed a visible normality. Having these in mind, different models were chosen to observe the behaviour of the data given the type of algorithm.

Supervised Learning models chosen for outcome prediction task are:

1. Linear Regression
2. Decision Tree Regression
3. Boosted Decision Trees
4. Support Vector Regression (SVR)
5. Neural Networks Regression
6. Gaussian Process Regression (GPR)

The datasets are standardized for all the models. For each of the models, the datasets are split into a 20% test set and 80% training and validation set. For validation, 5-fold cross validation is selected. The metrics for accuracy and performance used are Mean Squared Error, Root Mean Squared Error, Mean Absolute Error and R-squared.

4.1 Corner Indicators Dataset Output Prediction

Prediction models are built using regression algorithms to assess the vehicle performance on different driving styles, using the indicators of performance and style on every corner. The three output variables are chosen depending on domain knowledge and consultations with the team. These 3 variables are results and reactions of the vehicle, decided or happening upon the input actions of the driver and certain vehicle dynamics factors.

Grip Factor and Understeer Groups' variables have high correlation and this phenomenon can lower the performance of regression models. For each of the 3 output models, Trail Braking and Cornering variables are eliminated. Also state variables such as "Lap Reset", "Driver" and "Stint Number" are removed because these variables give exact information about corner time.

Model Number	*RMSE*	*MSE*	*MAE*	*R − squared*
1	0.15	0.02	0.11	0.93
2	0.24	0.06	0.13	0.82
3	0.25	0.05	0.16	0.83
4	0.52	0.27	0.43	0.17
5	0.16	0.02	0.08	0.92
6	**0.14**	**0.02**	**0.07**	**0.94**

Table 4.1: Results of prediction models for Corner Time.

Model Number	*RMSE*	*MSE*	*MAE*	*R − squared*
1	0.07	0.005	0.05	0.84
2	0.09	0.009	0.07	0.69
3	0.10	0.01	0.08	0.64
4	0.17	0.03	0.13	0.06
5	0.08	0.007	0.06	0.77
6	**0.05**	**0.003**	**0.04**	**0.90**

Table 4.2: Results of prediction models for Grip Factor.

Model Number	*RMSE*	*MSE*	*MAE*	*R − squared*
1	0.3	0.007	0.2	0.96
2	0.31	0.10	0.23	0.94
3	0.42	0.18	0.33	0.90
4	1.16	1.32	0.92	0.26
5	0.46	0.21	0.24	0.88
6	**0.21**	**0.04**	**0.14**	**0.98**

Table 4.3: Results of prediction models for Understeer.

4.1.1 Corner Time

For ease of explanation and simplicity, the best and worst performing models are selected to include in visual comparison.

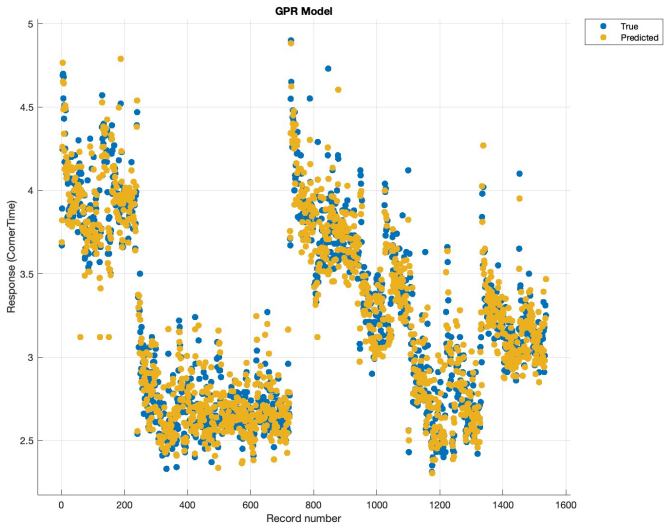

Figure 4.1: Predicted vs. ground truth points for GPR model.

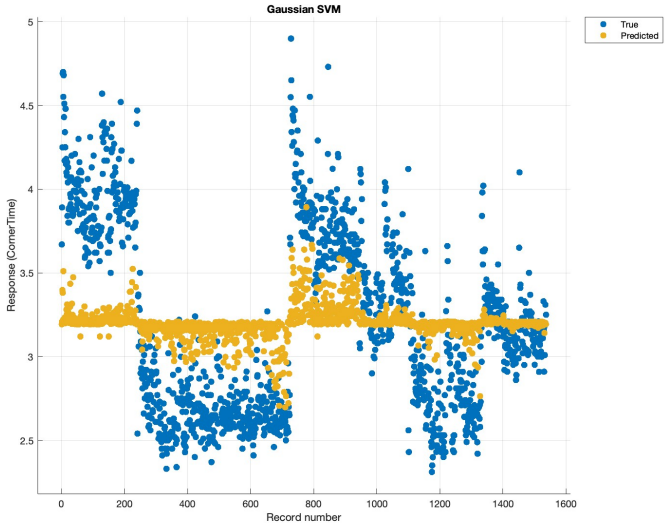

Figure 4.2: Predicted vs. ground truth points for SVR model.

4.1.2 Grip Factor

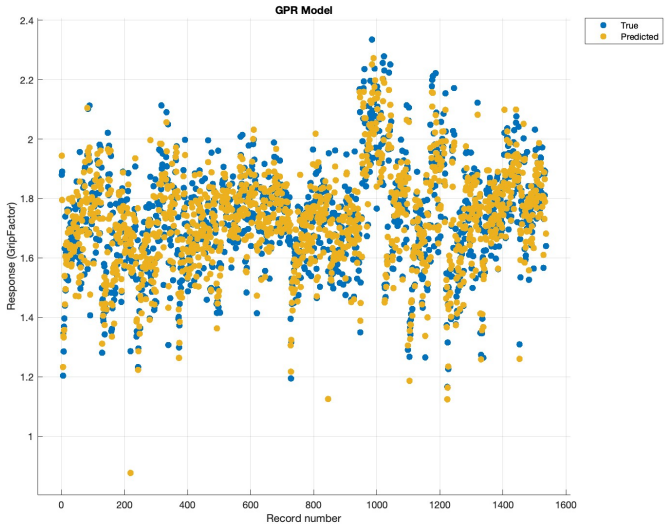

Figure 4.3: Predicted vs. ground truth points for GPR model.

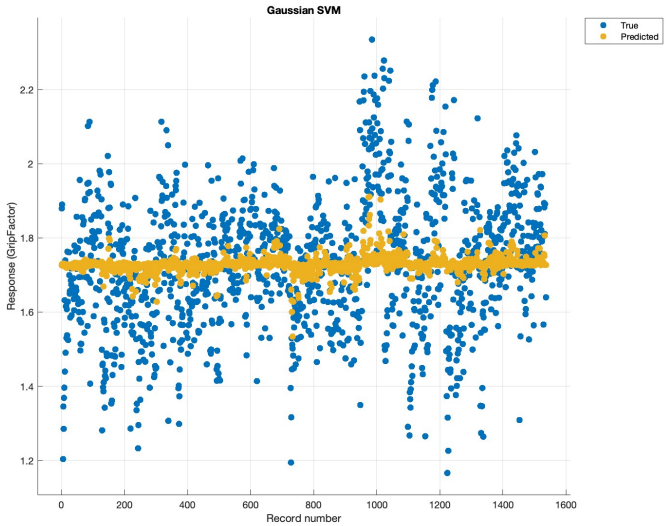

Figure 4.4: Predicted vs. ground truth points for SVR model.

4.1.3 Understeer

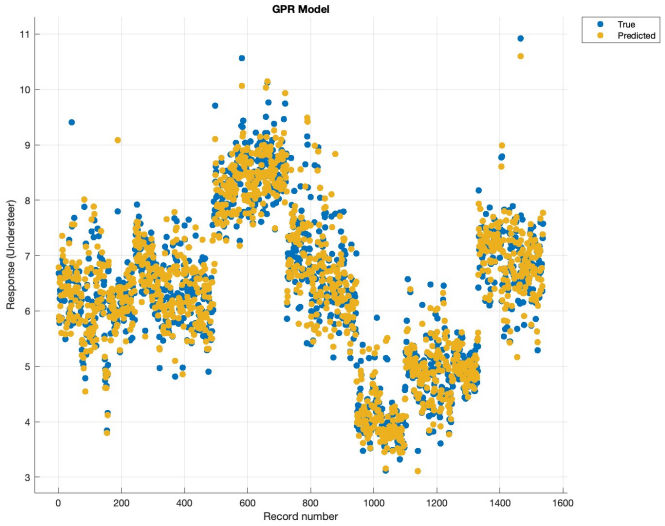

Figure 4.5: Predicted vs. ground truth points for GPR model.

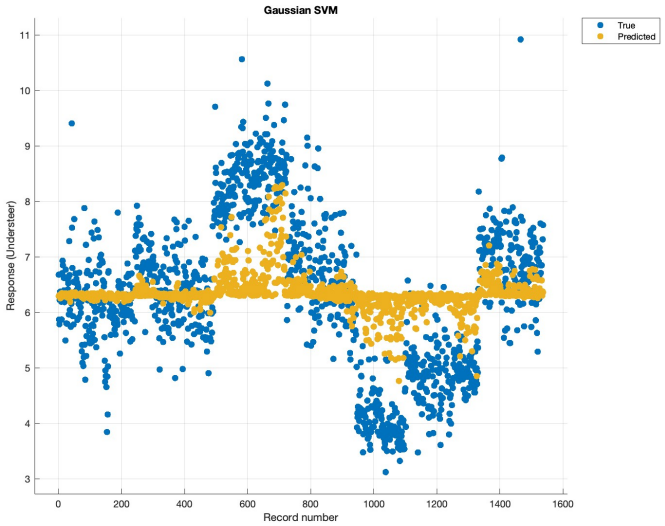

Figure 4.6: Predicted vs. ground truth points for SVR model.

Discussion of Results

Gaussian Process Regression models evidently outperform all models for each of the 3 outputs. SVR model results with the lowest performance, and the reason is suspected to be the outliers that may have damaged the predictive capabilities, or the usage of wrong values for the model hyperparameters. SVR has a more rigid working principle than GPR, given that GPR can model the data and the underlying uncertainties better due to its probabilistic nature. SVR works by trying to penalize the deviations to maximize the margin, therefore outliers can be more impactful on this model than GPR. Although the datasets were standardized before inference, the predicted points belonging to the SVR models appear as though the ground truth points are on a different scale than the predicted ones. This can be attributed to the unfitting hyperparameter selection, such as having too high or too low C, γ, and ϵ values. Gaussian Process Regression and Linear Regression perform similarly despite having different working principles. In the dataset there must be some features and relationships not entirely explainable by simple linear relationships and the use of kernels within the GPR model helped capture the margin of probabilistic and non-linear relationships better.

4.2 Simulator Signals Dataset Output Prediction

Following the performance of the models of the Corner Indicators Dataset, applying regression to the unfiltered simulator signals provided additional insights into vehicle dynamics. Output variables to be predicted were chosen as Vehicle Speed, Longitudinal and Lateral Acceleration after consulting the team. The input features only consisted of the three driver control input variables and numerical state variables. By using these limited inputs, the regression models aimed to capture complex vehicle dynamics, offering a better understanding of driving style effects and responses of the vehicle, despite the reduced feature set.

An important consideration was made not the include "YawRate" and "YawAngle" because of the high level of correlation that was discovered between these two variables and "LapTime".

Model Number	$RMSE$	MSE	MAE	$R-squared$
1	3.38	11.48	2.55	0.59
2	1.48	2.19	0.84	0.92
3	2.46	6.08	1.92	0.78
4	1.75	3.07	1.12	0.89
5	1.79	3.22	1.29	0.88
6	**0.68**	**0.47**	**0.21**	**0.98**

Table 4.4: Results of prediction models for Speed.

Model Number	$RMSE$	MSE	MAE	$R-squared$
1	0.23	0.05	0.18	0.91
2	0.15	0.02	0.10	0.96
3	0.16	0.02	0.11	0.95
4	0.15	0.02	0.10	0.96
5	0.15	0.02	0.11	0.96
6	**0.15**	**0.01**	**0.10**	**0.97**

Table 4.5: Results of prediction models for Longitudinal Acceleration.

Model Number	$RMSE$	MSE	MAE	$R-squared$
1	1.15	1.33	0.90	0.49
2	0.90	0.80	0.59	0.69
3	0.89	0.79	0.64	0.69
4	0.91	0.84	0.58	0.68
5	0.89	0.79	0.61	0.69
6	**0.85**	**0.73**	**0.58**	**0.71**

Table 4.6: Results of prediction models for Lateral Acceleration.

4.2.1 Vehicle Speed

For ease of explanation and simplicity, the best and worst performing models are selected to include in visual comparison.

Figure 4.7: Predicted vs. ground truth points for GPR model.

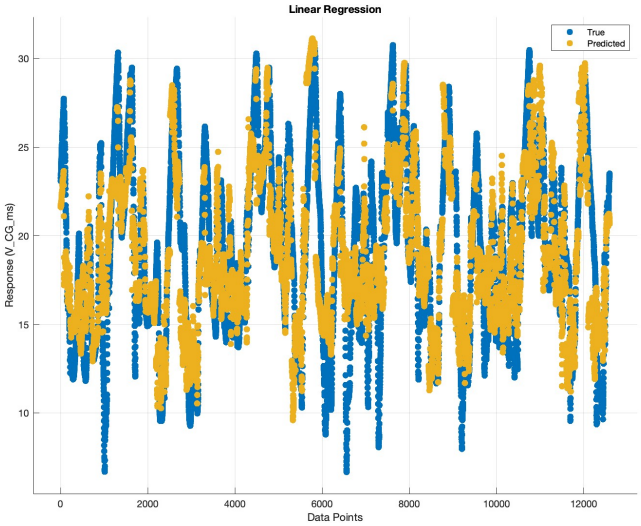

Figure 4.8: Predicted vs. ground truth points for LR model.

4.2.2 Longitudinal Acceleration

Figure 4.9: Predicted vs. ground truth points for GPR model.

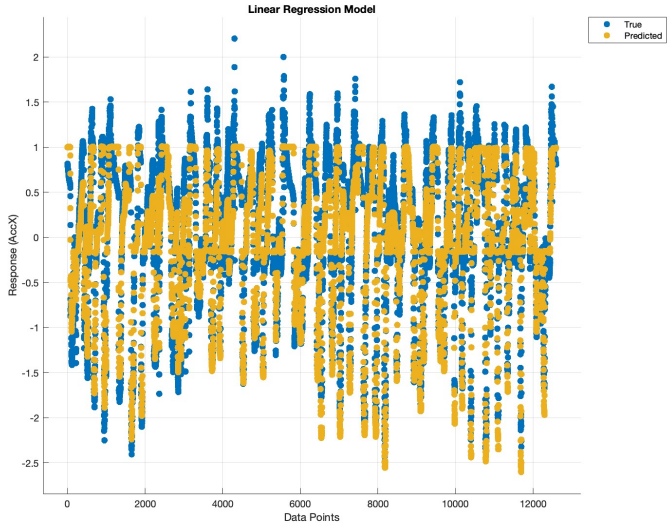

Figure 4.10: Predicted vs. ground truth points for LR model.

4.2.3 Lateral Acceleration

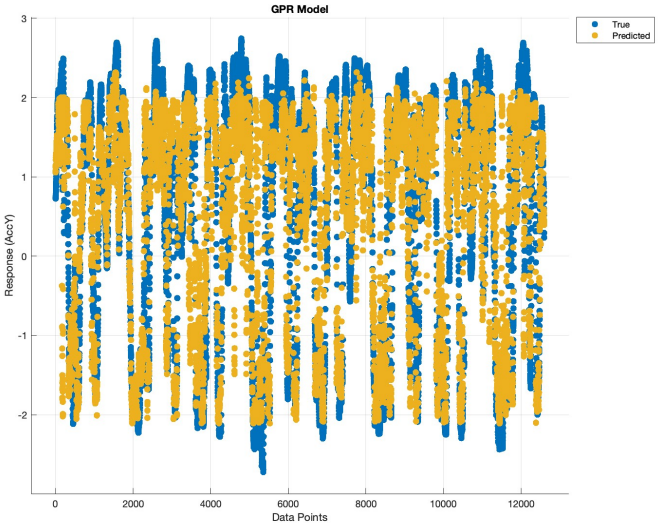

Figure 4.11: Predicted vs. ground truth points for GPR model.

Figure 4.12: Predicted vs. ground truth points for LR model.

Discussion of Results

Gaussian Process Regression (GPR) emerged as the best performing model also for this dataset. This was due to the model's ability to effectively capture both linear and non-linear relationships, handle small datasets, and model uncertainty in the predictions. Linear Regression performed the worse of all models, hinting at a more non-linear data outlook than Corner Indicators Dataset. Speed and Longitudinal Acceleration are seen to be more explainable than Lateral Acceleration. This finding suggests that Lateral Acceleration itself is very random and although "SteeringAngle" must be presenting some information to the model, it is not enough. Longitudinal Acceleration is more straightforward to explain, given that it is only dependent on braking and accelerating. As a result, regression models showed strong predictive capabilities even with limited amount of information in predicting Speed, Long. & Lat. Acceleration.

Chapter 5

Driving Style Analysis

In this section, the usages of ML models and data visualization techniques will be explored to test the hypothesis of the existence of individual driving styles and to accurately classify drivers based on the data from Corner Indicators Dataset. In Figure 5.1, the indicators used in analyzing driving style is given. These indicators are constructed and used following the framework made possible by Jorge Segers. [45].

	Performance	**Smoothness**	**Response**	**Consistency**
Acceleration	Throttle speed Full throttle time	Throttle erraticness	Full throttle point Time between throttle and braking	Evaluate performance for different corners
Braking	Peak brake pressure Braking length	Braking aggression Brake erraticness	Braking point Brake speed	Evaluate performance for different corners
Steering	Driving line variance Steering angle fixes	Steering aggression Steering erraticness	Steering speed Peak steering angle	Evaluate performance for different corners

Table 5.1: Driver evaluation indicators used in the Driving Style Analysis framework.

This analysis enables a deeper understanding of how different drivers interact with the vehicle under the same conditions, resulting in insights that can further personalize strategies and feedback, optimize car setups specifically for each driver, or predict performance based on which style is being demonstrated. Squadra Corse PoliTo had ongoing simulation test while this thesis was being written, as a result of this new drivers were added to the team during the 23/24 season. This is the reason of differing driver representation for different tasks.

For Formula Student and motorsports in general, identifying drivers by their unique driving style is pivotal for performance optimization and personalized feedback.

5.1 Driver Control Inputs Analysis

MATLAB environment is utilized as a powerful visualization and data analysis tool. This visual analysis serves as the basis for the following step, where Machine Learning models will be applied to quantitatively evaluate driving style and classify each driver by the style indicators.

Braking Action

Braking action is the application of pressure on the brake pedal by the driver to decelerate the vehicle. Braking performance can be measured by analyzing how smooth, fast and clean the deceleration is completed. Total braking distance and time spent decreasing speed should be kept to a minimum, since the end goal is to achieve fastest lap time. The smoothness of braking is crucial since braking erratically or in a messy way can cause the vehicle to loose balance.

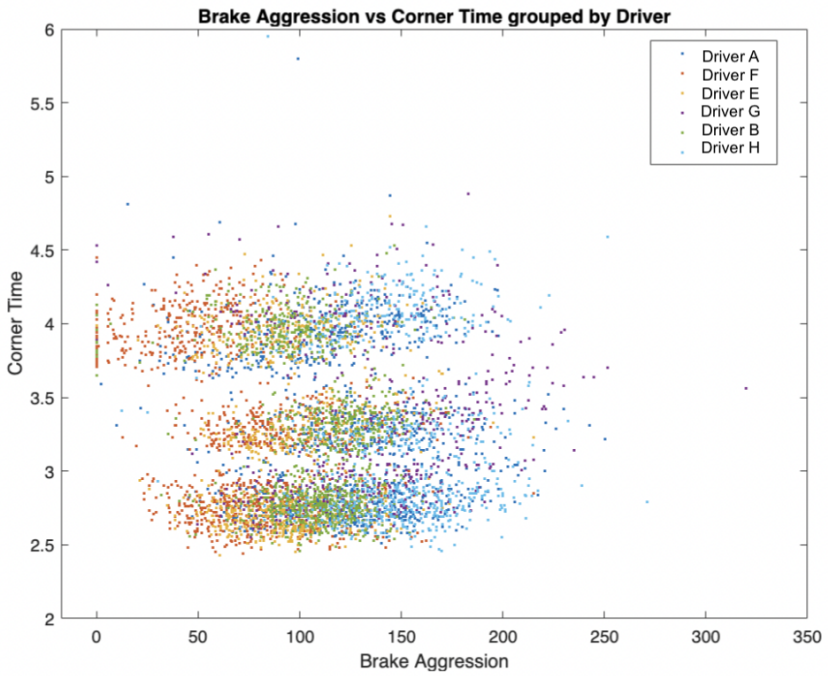

Figure 5.1: Brake Aggression and Corner Time per driver.

The 3 horizontal data clusters apparent in Figure 5.1 are representative of 3 groups of corners with different corner times. Claiming that there exists a straightforward relationship between "Brake Aggression" and corner time is evidently

a mistake. However, brake aggression of drivers and the variances between each of them are visible.

- Drivers F and E exhibit consistent and low brake aggression for every corner
- Drivers G and H have high brake aggression and more dispersed data points
- Driver B is consistent in his behaviour and has mid-level aggression
- Driver A exhibit closer-to-high brake aggression and inconsistent behaviour

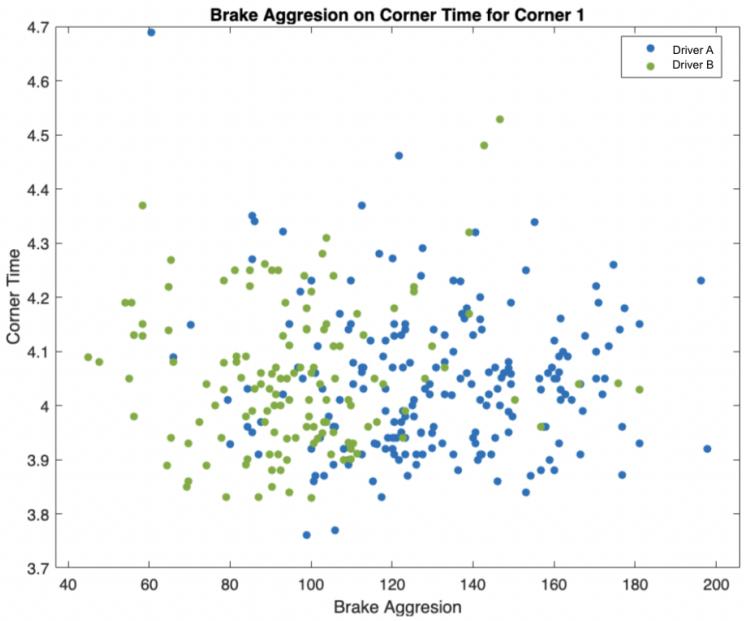

Figure 5.2: Comparison of Brake Aggression for Drivers A and B on Corner 1.

Supporting the previous visualizations, in Figure 5.2, Driver B exhibits lower brake aggression than Driver A. Brake Erraticness vs. Corner Time data points are also clustered in 3 groups representing the different corner times. Drivers F, H and E demonstrate consistent and low erraticness. Drivers A, B and G show dispersed and higher erraticness. In Figure 5.4, the previous statement is supported again. Driver B demonstrates the most consistent style along his stints. All of the six drivers seems to have lowered their erraticness values by the ends of their stints, learning and adapting to the vehicle.

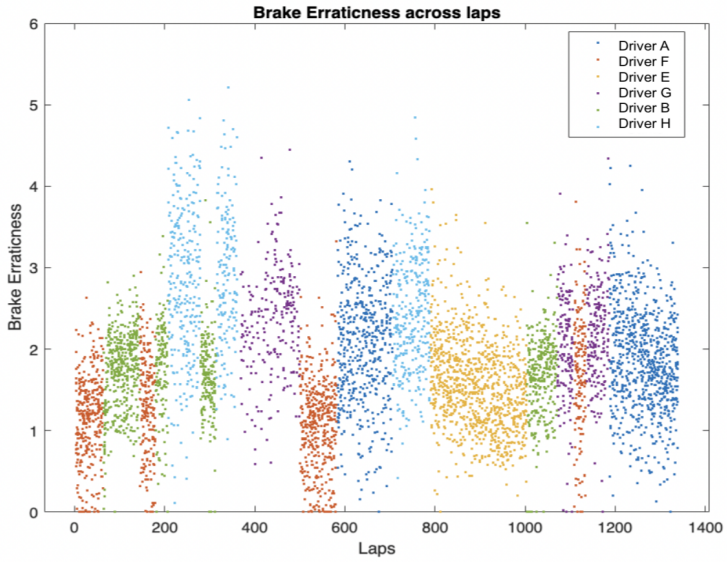

Figure 5.3: Brake Erraticness data points per driver.

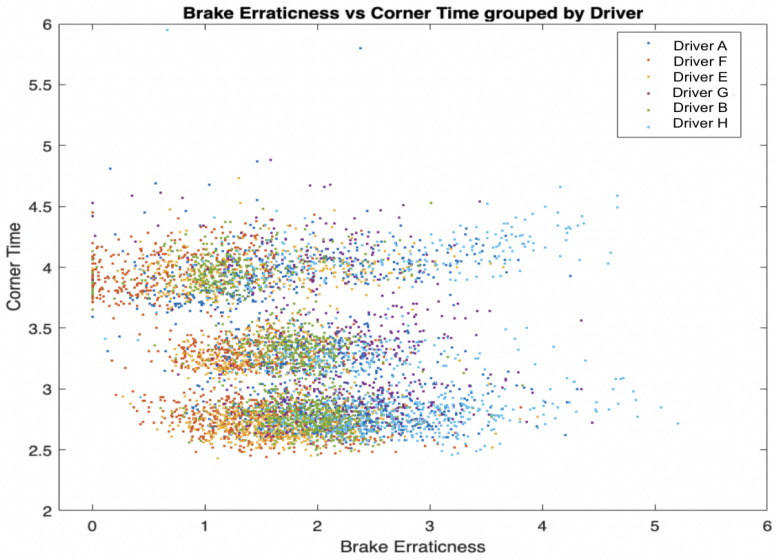

Figure 5.4: Brake Erraticness and Corner Time per driver.

Acceleration Action

Acceleration action is the pushing of the throttle pedal by the driver to increase the vehicle's speed. Acceleration performance can be evaluated by analyzing how smooth, quick, and controlled the increase in speed is executed. Just like braking, the execution of acceleration is crucial for reducing corner time. The style in which throttle is applied plays a part in the overall performance as well, since erratic or inconsistent throttle inputs can disrupt the balance of the vehicle, leading to loss of control or reduced grip, particularly in high-speed corners. Smooth and precise acceleration application not only maximizes speed but also helps to maintain stability and prevent understeer.

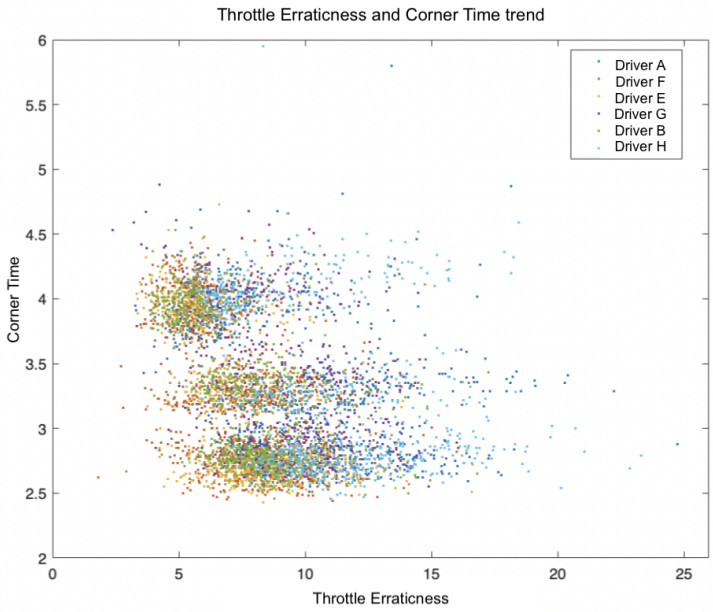

Figure 5.5: Throttle Erraticness and Corner Time per driver.

In Figure 5.5, Throttle Erraticness trends are shown with the respective corner times. Drivers B, F and E's data points demonstrate lower levels of erratic behaviour and are densely clustered. Drivers A, G and H's data points demonstrate a more erratic style and higher levels of variance.

Figures 5.6 and 5.7 feature a close-up view of throttle signals of Drivers A and B on Corners 2, 4 and 5. These corners were selected after analyzing all corners for visible variance in behaviour. Driver A actuates the throttle pedal in a less homogeneous manner, featuring fluctuations in his throttle signal trace. Driver B applies throttle pressure swiftly and in a clean manner, resulting in less erratic and more efficient acceleration.

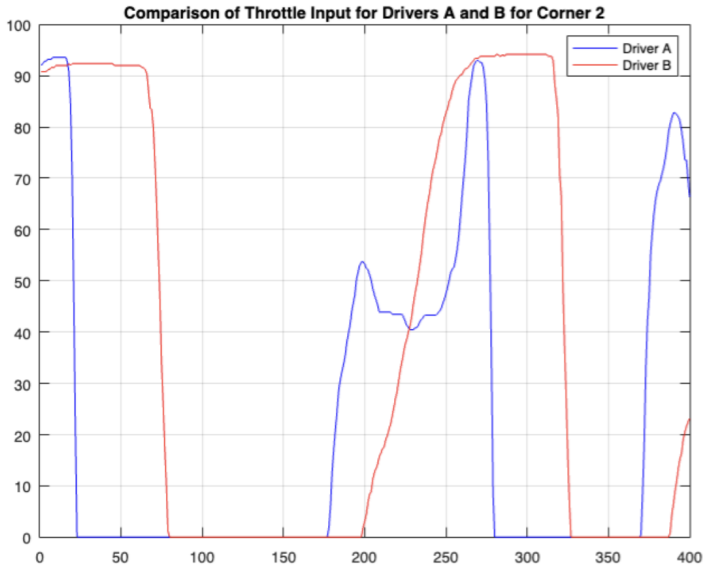

Figure 5.6: Throttle signal comparison on Corner 2.

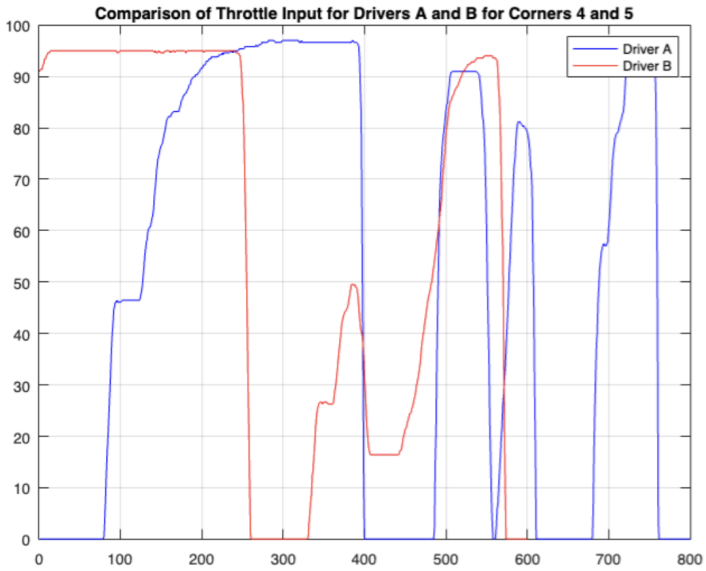

Figure 5.7: Throttle signal comparison on Corners 4 and 5.

Steering Action

Steering action represents turning of the steering wheel by the driver in order to change the direction of the vehicle. Steering performance is defined as smoothness, precision and the quickness of response. The quality of steering is key for successful cornering and fast corner exit. If the steering wheel is turned in excessive angles very suddenly the balance of the vehicle can be damaged and lead to understeer, oversteer or even loss of control of the vehicle. Timing of steering is also crucial, entry point of a corner and the line taken by the driver in the track are what determines the overall performance.

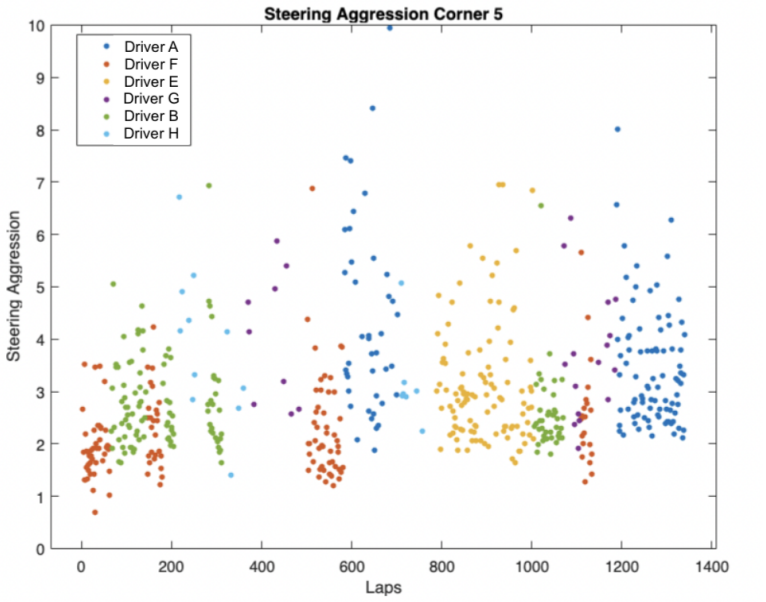

Figure 5.8: Steering Aggression data points for Corner 5.

In Figure 5.8 it is seen that the general Steering Aggression style is similar for the drivers on Corner 5, with Driver F having a lower than average style and Driver A exhibiting more heterogeneous behaviour than the rest of the drivers.

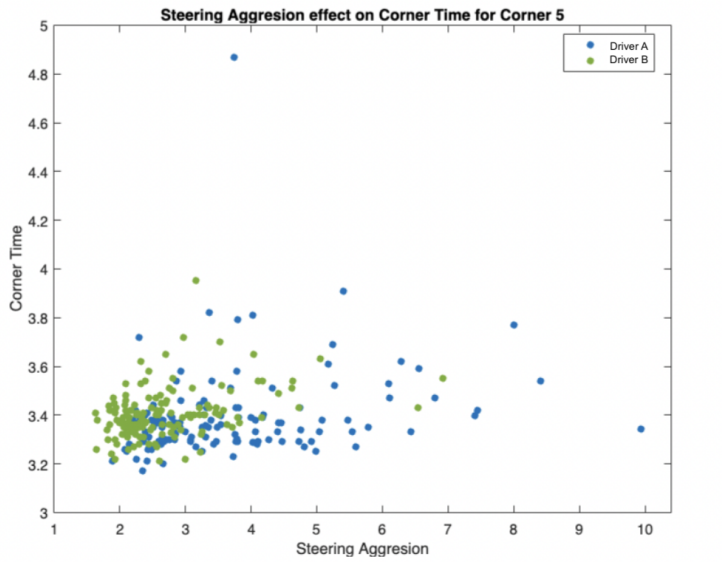

Figure 5.9: Steering Aggression and Corner Time relationship for Corner 5.

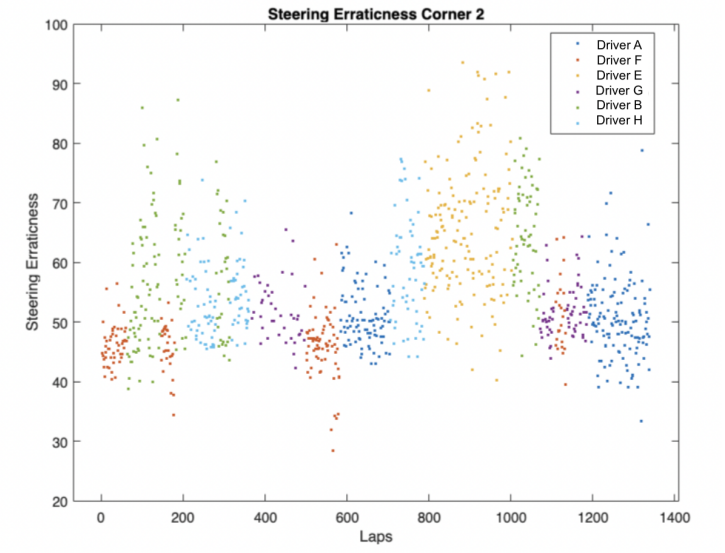

Figure 5.10: Steering Erraticness on Corner 5.

The comparison of Drivers A and B in Figure 5.9 shows Driver A's data points are more dispersed than Driver B's and Driver B demonstrated a less aggressive steering style.

Driving Line and Signal Trace Analysis

Driving Line is the path the vehicle follows through a track. The optimal driving line maximizes speed and minimizes lap time by balancing the cornering deceleration and the acceleration. However, deciding on the best line for a corner depends on the corners before and after, as noted by Segers in [45]. The purpose of a driver is to take the smoothest and fastest line, maintaining speed through the corner and exit with highest possible speed.

Figure 5.11: Driving line comparison for Driver A and B.

Driving lines of Drivers A and B are compared in Figure 5.11 by overlaying the positional signals of the vehicle on the track. It is challenging to offer an accurate comparison of driving styles by simply focusing on the driving line of a single lap, however differences are still visible. Driver A usually followed a smaller corner radius path hence increasing the road travelled. Driver B is rotating the vehicle more in corner entry and exit sections, which could suggest opposing styles. Checking also Figure 5.13, it can be concluded that Driver B was driving the vehicle to its limits whilst preserving speed. In Figure 5.12, it can be noted that Driver A decelerates and accelerates more frequently than Driver B, suggesting a more erratic throttle and brake application. Driver B doesn't slow down the vehicle completely for corners as Driver A does, and uses the pedals more smoothly, demonstrating a less erratic driving style.

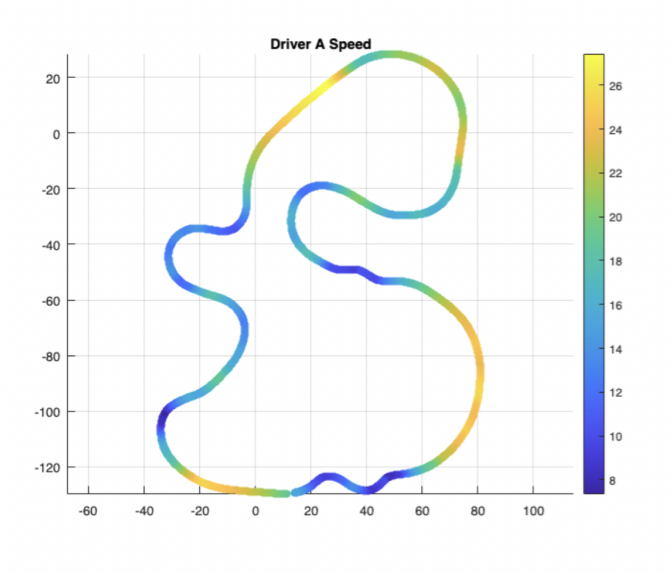

Figure 5.12: Speed of Driver A on track layout.

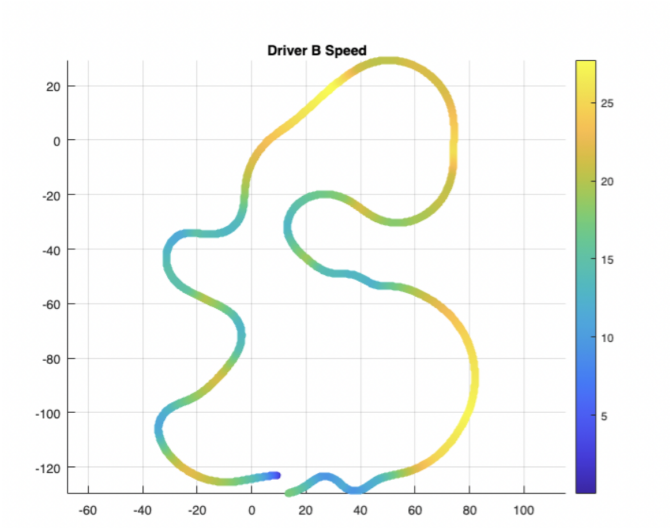

Figure 5.13: Speed of Driver B on track layout.

From Figure 5.15 it is visible that Driver B achieves a style closer to the desired one for braking action. Driver A has a higher peak brake signal throughout his lap compared to Driver B. This means Driver A brakes harder at every corner entry, exhibiting a more aggressive behaviour. It is also seen in Figure 5.14 that Driver A tends to apply repetitive pressure on the brake pedal anticipating the corner rather

Figure 5.14: Braking zones of Driver A on track layout.

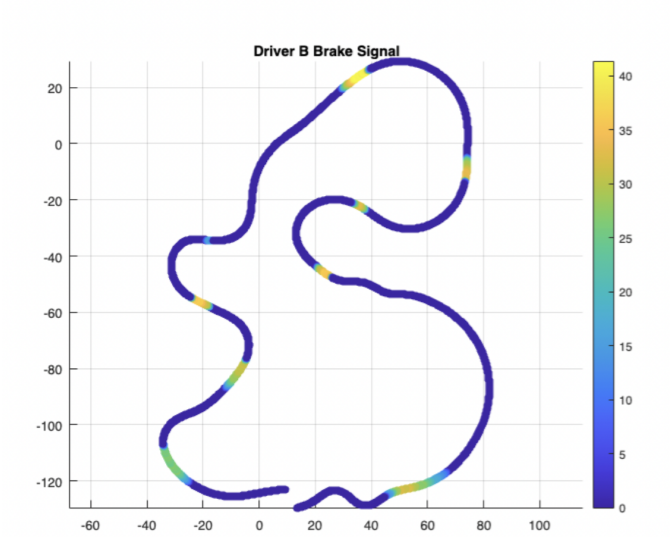

Figure 5.15: Braking zones of Driver B on track layout.

than a smooth and homogeneous application.

In Figures 5.16 and 5.17 throttle signal of Drivers A and B are compared. Driver A shows a more erratic throttle application behaviour than Driver B. There exists small and frequent applications of throttle pressure approaching or exiting a corner for Driver A. Driver B applies throttle in a consistent manner, without lifting the pedal. This style is healthier for the vehicle, and generally results in lower lap time.

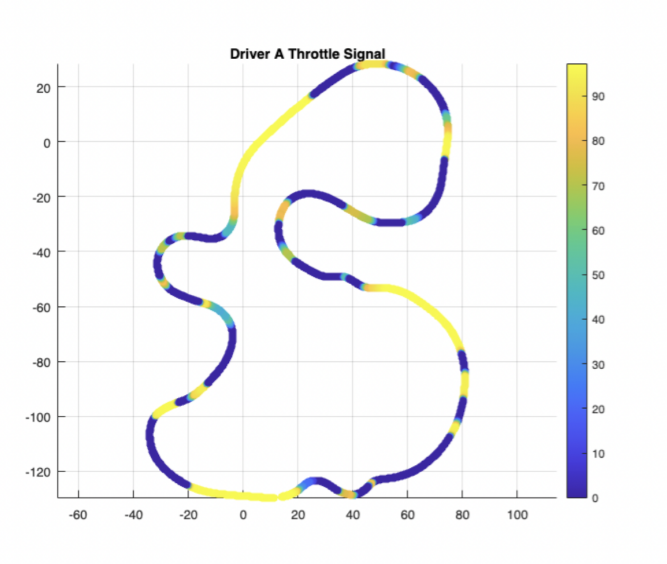

Figure 5.16: Acceleration zones of Driver A on track layout.

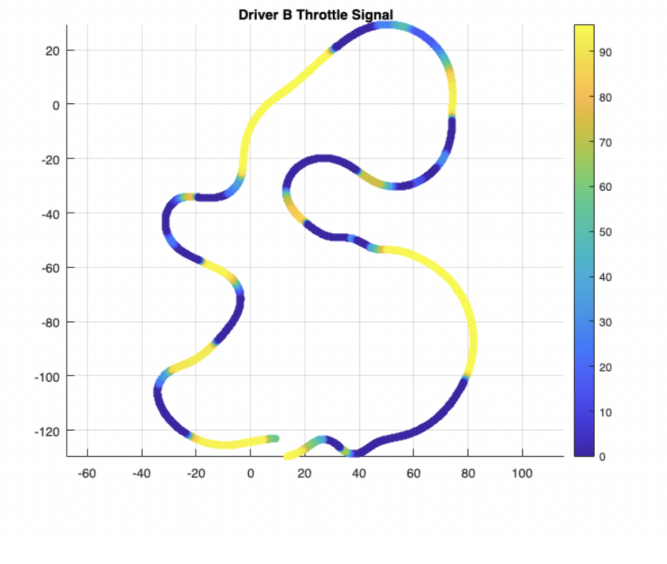

Figure 5.17: Acceleration zones of Driver B on track layout.

Main reason for this is time spent accelerating essentially means increasing speed, hence the more time a driver spends in full throttle the faster his lap time will be.

In Figures 5.18 and 5.19, green and yellow symbolize when the driver is turning left, whilst shades of blue represent turning right. It can be seen there exists abrupt changes of color in Driver A's signal trace pointing to a more erratic style

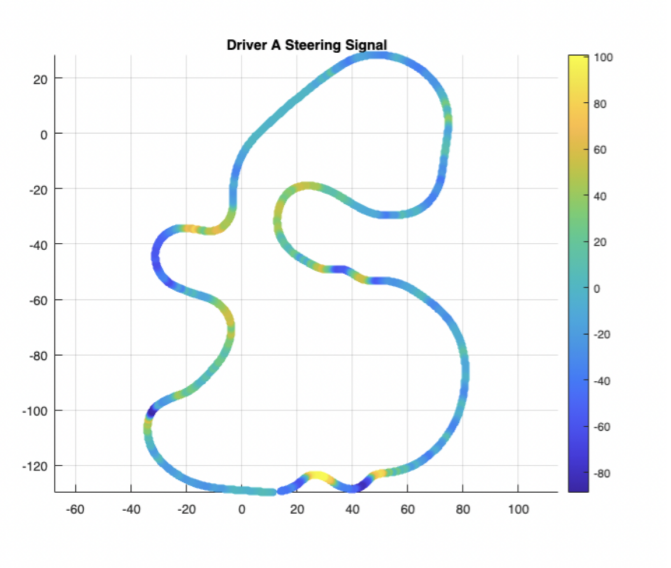

Figure 5.18: Steering behaviour of Driver A on track layout.

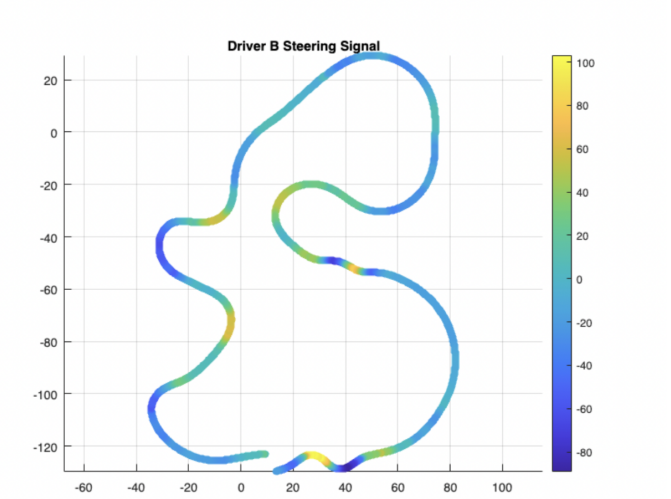

Figure 5.19: Steering behaviour of Driver B on track layout.

of steering. The sudden changes in steering angle is considered erratic, even if these actions are taken to correct the vehicles direction in a corner. Driver B shows consistently increasing and decreasing color differences in his steering angle signal trace. This corresponds to a smoother and less erratic steering style and should help with the overall balance and performance of the vehicle.

5.2 Identifying Drivers by Supervised Learning Models

After deploying visualization tools to uncover the inherent behavioral patterns of the SCP drivers, it was determined that there exists certain driving style features to identify each driver by. To further test and prove this hypothesis, a "Driver Identification" framework was developed by using Supervised Learning models, namely Decision Trees, Support Vector Machines and Neural Networks. These models provide a robust framework to map driving inputs to drivers, and to learn behavioral patterns, therefore allowing an accurate identification.

ANOVA test is conducted for feature selection and understanding the scale of the features impacts. ANOVA tests' results represent the amount of variance between the means of multiple groups. A significant result in ANOVA test indicates the feature in question has considerable impact on differentiating between drivers. The higher the score of a feature, the more it helps distinguishing one group (driver) from another.

It was seen that the top 10 most important features for identifying drivers are:

1. Brake Erraticness
2. Throttle Erraticness
3. Peak Brake
4. Brake Aggression
5. Brake Speed
6. Throttle Speed
7. Steering Speed
8. Steering Aggression
9. Braking Point
10. Turn-In Point

According to the results of the ANOVA test, braking actions group was identified as the most distinguishing set of variables. More specifically, "Brake Erraticness" is the key variable to identify drivers, and this supports the findings of the visualizations. "Throttle Erraticness" follows closely, implying that irregularities in throttle pedal actuation also play a role in distinguishing drivers. "Erraticness" is

an indicator of the irregularities and sudden changes of the drivers actions, and evidently this factor differs significantly for each driver.

The structure of the models are:

Output Variable:	"Driver"
Input Variables:	Output Groups Variables and Brake, Steering Throttle and Position Indicator Groups

Model Selection and Deployment

Each model was chosen for its capability in handling the complexity and nature of the dataset, while also creating a balance between accuracy, interpretability, and computational efficiency. By comparing the performance of these 3 models, the purpose is to test which is more adaptable to the specific characteristics of driver classification problem. Training set is 80% and test set is taken as the remaining 20% of the dataset. Accuracy, Precision, Recall and F1-score are selected as metrics of performance.

- **Decision Tree** model was created with Gini Impurity Index as the splitting decision with maximum depth of the tree set to a low number to limit the complexity.
- **SVM** model followed a One-vs-Rest approach. In this approach, SVM model creates a classifier for each class, resulting in 6 classifiers for 6 drivers. This allows the model to map differences of the classes. An RBF kernel was used considering the non-linear nature of the dataset.
- **Neural Network** model consisted of 3 hidden layers with ReLU activation function, 6 nodes in the output layer for each driver with softmax activation for multiclass classification. Categorical Cross-Entropy loss function was used with Adam optimizer.

Results

Metric	**Decision Tree**	**Neural Network**	**Support Vector Machine**
Accuracy	0.5373	0.7742	0.8066
Precision	0.5384	0.7758	0.8147
Recall	0.5190	0.7718	0.7985
F1-Score	0.5285	0.7738	**0.8065**

Table 5.2: Performance comparison for DT, NN, and SVM Models

SVM outperforms the other two models across all metrics and the other models failed to capture the niche driving behaviors that differentiate drivers. This finding suggests that the non-linear and complex relationships between input variables and drivers identity was captured the best by SVM architecture that allows classifying non-linearly separable groups by the RBF kernel.

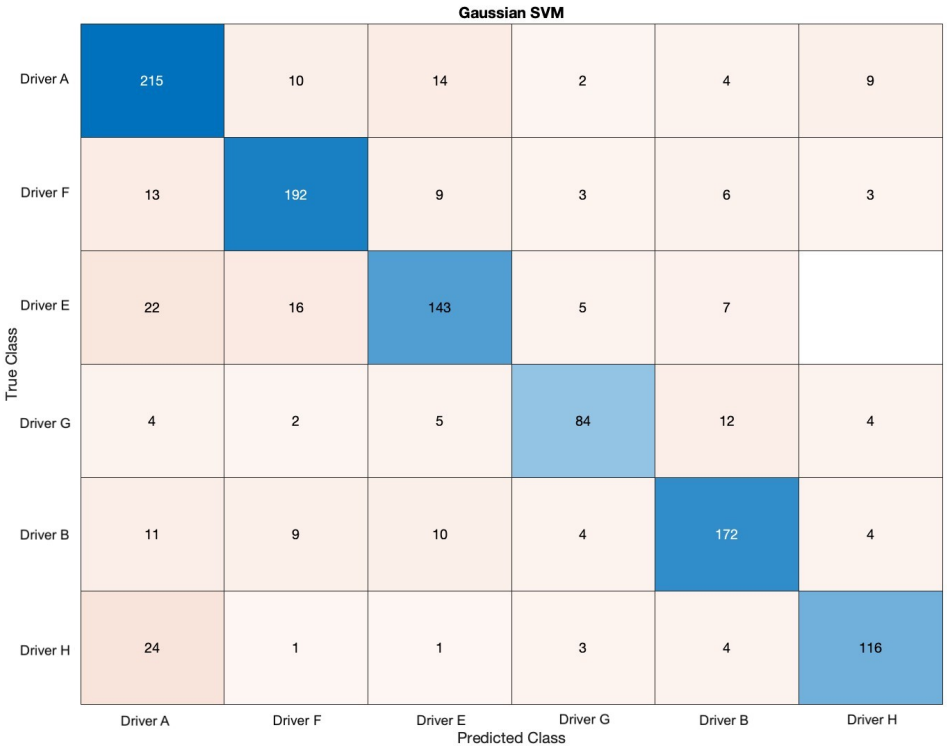

Figure 5.20: Results of SVM model.

Another interesting result is the differences of the accuracy metric for each driver. Driver F has the highest number of correct predictions in all 3 models, followed by Driver A and Driver B. This finding also relates and supports the results of the previous step, as it was seen that the style of these drivers were more apparent and different compared to the others. By backing up the claim made by visualization tools, the existence of better-performing driving styles can be underlined.

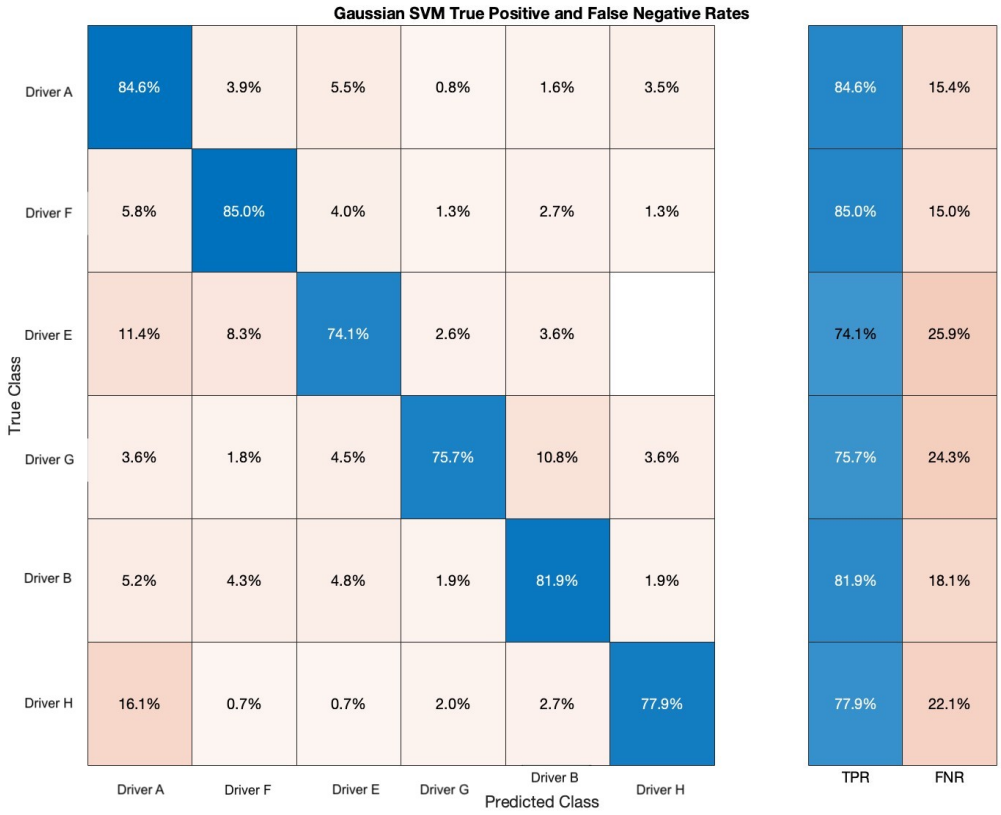

Figure 5.21: True Positive and False Negative rates of SVM model.

In the best performing SVM model, Driver E and Driver H were commonly misclassified as Driver A and Driver G was misclassified as Driver B. This result shown in Figure 5.20 and Figure 5.21 highlights a similar style adapted by these respective drivers that the model struggled to tell apart from each other.

Chapter 6

Strategy Optimization

This chapter delves into optimization of driving style and strategy. Two approaches are followed and compared to have a better understanding of the characteristics of the specific context and to reach the best results possible: Numerical Optimization and Imitation Learning.

First part of this chapter is dedicated to optimization and its application to the problem and the results. Optimization techniques are powerful for a wide range of problems, however they can struggle with capturing nuanced relationships and need thorough detailing of the problem space. In the context of strategy optimization for Squadra Corse PoliTo, the objective function to optimize, physical bounds of vehicle dynamics phenomena, the vehicle model and trajectory definition is missing. This lack of realization can be attributed to the gaps in research and opportunity for conducting a deep analysis on the FS vehicles and drivers. With no objective function or sufficient information on the environment and search space, modelling of the Optimization model was expected to create an obstacle. Considering these limitations, Imitation Learning was selected to demonstrate how supervised learning algorithms can be used as an alternative tool to optimization methods, and to create a driver model that can make decisions for new scenarios by itself.

6.1 Numerical Optimization

Traditional optimization methods are mathematical models that tries to reach an optimal solution for maximization or minimization problems. Having understood this, expecting these methods to fully capture the randomness of human behaviour and physical phenomena is not justified. Driving styles and the factors that play into the randomness of drivers actions were highlighted in Chapter 2, with a detailed analysis of the specific styles in Chapters 3 and 5. It was seen that not only there exists infinite differences between the actions of diverse individuals, but

also between the consecutive actions of the same individual.

Optimization algorithms require a definition of the objective by modelling the variables and their respective coefficients as a function of the relationships. In this case however there exists no such objective function for the output variables of the datasets. A solution for this obstacle was designed by only repeating a part of the "Outcome Prediction" step; Linear Regression. This algorithm works by learning the regression line formula that fits the dataset, linking the output variable with the input features. Therefore, by applying Linear Regression, the regression line formula for the output variable was obtained.

After completing the data prerocessing step and standardizing the dataset, Linear Regression Models were tested to predict "CornerTime" for Corner Indicators Dataset. Simulator Signals Dataset wasn't used in this step given that in Chapter 4 it was seen that Linear Regression model didn't perform in a satisfactory level when deployed to predict Simulator Signals Dataset's output variables.

Another limitation for this task was the number of input features in the dataset. Optimization can be used for multivariate problems however inputting more than 20 or even possibly 30 features isn't possible. To remedy this, a subset of features was identified for Corner Indicators Dataset. Feature selection was conducted by using the "F-test" to identify the 10 most influential features to include in the objective function to predict "CornerTime".

The selected features are:

1. MinimumSpeed
2. ExitSpeed
3. PeakCurvature
4. BrakingPoint
5. ThrottleErraticness
6. ThrottleSpeed
7. BrakeSpeed
8. SteeringSpeed
9. ThrottleIntegral
10. SteeringIntegral

Linear Regression Model's Mean Squared Error was found to be 0.097 with $\mathbb{R}^2 = 0.68$. The predictive capability and explainability of the output variable by the input features is evidently lower compared to the results of the "Outcome Prediction" task in Chapter 4. This can be explained by the lack of data input for the model in this task, 10 input features may not have given the model enough information to accurately predict "CornerTime". Also the relationships between input features and "CornerTime" was proved to be non-linear and highly complex in the previous chapters. However it was decided that sacrificing a margin of accuracy to obtain an objective function was justified in order to move forward with optimization. By using the formula given below and the input features, 68% of the variance in "CornerTime" can be explained.

$$\begin{aligned} CornerTime = {} & 3.6849 + 0.0005 \cdot X_1(\text{MinSpeed}) - 0.0092 \cdot X_2(\text{ExitSpeed}) \\ & - 5.2751 \cdot X_3(\text{PeakCurvature}) + 0.0365 \cdot X_4(\text{BrakingPoint}) \\ & + 0.0152 \cdot X_5(\text{ThrottleErraticness}) - 1.5786 \cdot X_6(\text{ThrottleSpeed}) \\ & - 0.4044 \cdot X_7(\text{BrakeSpeed}) - 0.4291 \cdot X_8(\text{SteeringSpeed}) \\ & + 0.0001 \cdot X_9(\text{ThrottleIntegral}) + 0.0001 \cdot X_{10}(\text{SteeringIntegral}) \end{aligned}$$

The formula given above is used as input for the Gaussian Process Bayesian Optimization model. One point of consideration is the applicability of the results of the optimization model for Squadra Corse PoliTo. Driver control features that can be directly adapted and changed should be included in the objective function, since the main purpose of this section is to supply the team with feedback on performance optimization. For this reason, "PeakCurvature", "BrakingPoint", "ThrottleSpeed", "BrakeSpeed" and "SteeringSpeed" variables from the function are chosen to be optimized. To constrain the optimization algorithm search space, the bounds of each variable is calculated from the data by extracting the (Minimum, Maximum) range.

Variable	**Min**	**Max**	**Mode**
Peak Curvature	0.039517	0.232727	0.039517
Braking Point	0.000000	25.347358	4.065124
Throttle Speed	0.089898	1.184589	0.089898
Brake Speed	0.000000	0.433795	0.000000
Steering Speed	0.102369	1.103498	0.102369

Table 6.1: Statistical metrics for the decision variables.

After defining the bounds for the decision variables, the values to use for the 5 non-decision variables were extracted for each corner by using the findings from Chapter

5. Best "CornerTime" value for each corner was calculated and the driver inputs for that specific lap and corner were noted. When there were multiple instances of the best "CornerTime" value by different drivers, the one with the best style was chosen, as analyzed in "Driving Style Analysis". By extracting 7 different sets of fixed values for "MinimumSpeed", "ExitSpeed", "ThrottleErraticness", "ThrottleIntegral" and "SteeringIntegral", expert demonstration of each corner was selected.

Modelling the Optimization Problem

- **Objective function:**

$$\min z = f(X_3, X_4, X_6, X_7, X_8)$$

- **Decision Variables:**

$$X_3, X_4, X_6, X_7, X_8$$

- **Constraints:** Bounds of variables given in Figure 6.1

Corner	CornerTime	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_5}$	$\mathbf{X_9}$	$\mathbf{X_{10}}$	Driver
1	3.96	43.84	68.32	5.48	10537.57	8454.95	F
2	2.64	44.41	64.98	8.94	11620.14	7859.68	B
3	2.74	41.16	62.91	7.40	8987.03	15778.62	E
4	3.82	43.78	92.76	7.22	26396.77	8043.27	H
5	3.31	72.45	88.92	12.53	14823.06	4332.95	B
6	2.61	61.34	72.76	12.25	5747.20	5098.42	F
7	3.18	48.78	55.66	8.06	7195.82	11705.17	A

Table 6.2: Fixed values for non-decision variables.

Gaussian Process Bayesian Optimization takes the bounds of the search space and objective function as input besides the model hyperparameters. The most important aspect of GPBO is that the objective function is only used as a starting point that will be developed as the model works. The architecture for optimization is built to run GPBO for each set of fixed corner values, and to record the output values for each iteration of the loop. The objective function takes in the fixed values for the non-decision variables and optimizes the decision variables within the given bounds for each corner. At the end, 7 sets of input action values and 7 "CornerTime" values are found for each corner.

Results of optimization are:

1. For Corner 1, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 1.31s.

2. For Corner 2, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 1.44s.

3. For Corner 3, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 1.96s.

4. For Corner 4, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 2.66s.

5. For Corner 5, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 1.26s.

6. For Corner 6, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 0.57s.

7. For Corner 7, the optimal feature values are [0.23, 0.0, 1.18, 0.43, 1.10] and the optimal "CornerTime" value is found as 1.46s.

Despite multiple attempts, the model was unable to produce valid or meaningful results. This may be attributed to factors such as lack of information, insufficiency of data, or the inherent complexity of the problem not being captured by the objective function. The linear regression function was expected to perform as such, since there was a gap of explainability. The optimal driver actions are found to be the same for each corner, which means the model was not able to discern different corners. Having said this, it can also be noted that optimal values of corner time are lower than the actual data, therefore the model has produced a minimized result for the system. The values are not meaningful in a real-world setting because it would not be possible for the drivers to reach 1.30 seconds etc. Required additional information about the vehicle model, trajectory or track limits would've presented the model with a more realistic understanding of the problem domain. Considering that the model had little to no information about vehicle dynamics or historical data, the invalid optimal values for corner time are justified.

6.2 Imitation Learning

In this step, the final task for Formula Student vehicle performance optimization is implemented. Imitation Learning is chosen for the problem at hand given that compared to Reinforcement Learning, in IL there is no need to build an environment for the agent or to supply the reward and loss functions to enable learning policies

by trial and error process. The purpose of this section is to build a smart agent by using IL to imitate the expert drivers behaviour and style by learning the past actions.

Behavioral Cloning method - Imitation Learning workflow

1. Collect expert demonstration data with (state,action) pairs
2. Preprocess the data to extract relevant features and ensure data integrity
3. Train a Supervised Learning model to map states to actions (to learn a policy)
4. Evaluate the learned policy by deploying it in the target scenario

Dataset and Preprocessing

For Imitation Learning, one of the most important requirements is "expert demonstrations" which are data points taken from the exemplary actions of experts. These demonstrations are collected to ensure the high quality of actions learned and imitated by the agent. If there are non-expert actions in the data, the agent may learn these faulty behaviours and imitate them, resulting in a less than satisfactory model performance. Imitation Learning is used extensively in autonomous driving cases, hence the performance and accuracy of the imitated actions are crucial for the problem.

Simulator Signals Dataset was analysed for extracting driving styles in Chapter 5, and relating to these findings, the existence of better-performing driving styles is proved. Certain characteristics that drivers exhibit allow to better extract the potential of the vehicle, resulting in higher overall performance. Drivers who demonstrated an overall higher performance and expert-like style were chosen for this task. From the dataset, best lap times for Drivers B, F, G and H were found, then these 4 laps were merged to create "expert demonstrations".

To make sure the agent can learn a specific and detailed policy, the dataset was then split into different corners, resulting in 10 datasets for each corner in the SCP track. These datasets were then standardized as explained in Chapter 2 to ensure the integrity of inference. Imitation Learning requires the data to be presented as (state, action) pairs, in which the state features represent the conditions and factors surrounding the agent at every instance, and action variables are the actions the agent should imitate. Each dataset was split into state and action subsets, resulting in 20 datasets in total, two for each corner.

- **State Features** were chosen after consulting the Correlation Analysis results from Chapter 3, given that for IL, the features must be independent of each other. After removing the features that had high correlation values, the remaining subset was made from "AccX", "AccY", "PosX", "PosY", "V_CG_1", "YawAngle" and "YawRate". These features carry invaluable information on the exact state of the vehicle and driver at each time instance.
- **Action Variables** are chosen as the three driver input variables, "Brk_Input", "SteeringAngle" and "Throttle" to create a "driver model".

Model Architecture

Neural Networks were used for Behavioral Cloning and `BCNet` feed-forward network with fully connected layers was constructed for inference. The model architecture given in Figure 6.1 consists of an input layer with $X_n; n = (1, 2, 3, 4, 5, 6, 7)$ input features, 3 hidden layers each with ReLU activation function to introduce non-linearity to the model each and an output layer with $y_k; k = (1, 2, 3)$ representing the 3 driver actions to be mapped to a policy $x_n = \pi(y_k)$.

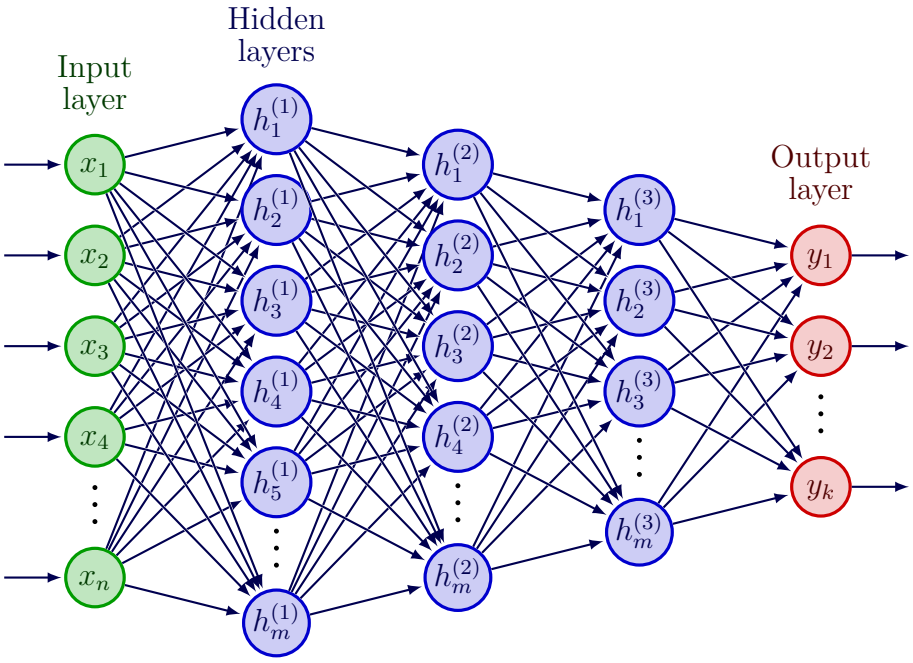

Figure 6.1: `BCNet` model architecture.

The explicit notation of the policy is $x_n = \pi(\tau, \beta, \varsigma)$, where x_n is the state features, τ represents "Throttle", β represents "Brk_Input" and ς represents "SteeringAngle".

In the output layer an activation function wasn't used given that the output values are continuous and can be inferred directly without further modification. "Adam" optimizer was implemented to minimize the models loss by updating the network weights.

Deployment and Results

In this section, the results of `BCNet` model's performance will be explained. The results are indicative of the model's capability to imitate expert actions, and it is calculated in terms of MSE (Mean Squared Error) for training and evaluation steps.

`BCNet` model was trained for 20 epochs to ensure convergence for each corner. The training loss decreased over each epoch for every corner model for 20 epochs, as shown in Table 6.3, indicating that the models were effectively learning from the expert demonstrations.

Corner	Training Loss	Test Loss
1	0.0336	0.0324
2	0.0319	0.0295
3	0.0700	0.0660
4	0.0533	0.0514
5	0.0742	0.0691
6	0.0961	0.0900
7	0.0529	0.0511
8	0.0323	0.0302
9	0.1234	0.1200
10	0.0277	0.0264

Table 6.3: Training Loss of BCNet at 20th epoch and Test Loss of each corner.

After training, the model was evaluated on a test set, which the model had not been trained on. Test losses are given in Table 6.3 and the results indicate that the models were able to generalize well and could imitate expert behaviour for each corner with high levels of accuracy. Model performed best for Corner 10 and worst for Corner 9, and to inspect the imitated actions better these corners were visualized in Figures 6.2 and 6.3.

Figures 6.2 and 6.3 show the predictive errors of the `BCNet` model. The model predicted higher steering angles and less brake input than the actual data for Corner 9, which indicates the presence of varying driving styles adapted for this corner. For

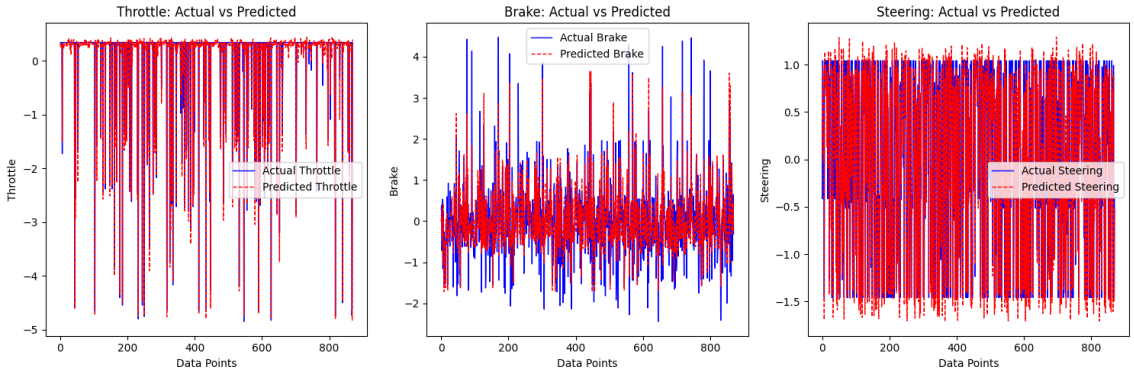

Figure 6.2: Comparison of actual and predicted actions for Corner 9.

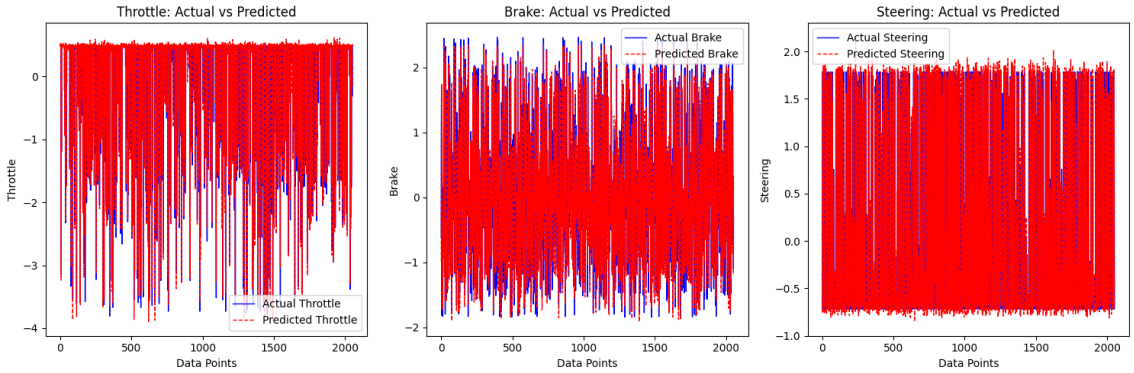

Figure 6.3: Comparison of actual and predicted actions for Corner 10.

Corner 10, the imitated behaviour almost matches the actual data of the drivers perfectly, meaning that this corner was a more adaptable and generic one. The `BCNet` models successfully learned to imitate expert behaviour with reasonable accuracy, as demonstrated by the decreasing loss during training and a low test loss during evaluation. The model is capable of capturing complex relationships between state features relating to the vehicle and the corresponding driver actions, proving the adaptability of Imitation Learning approach for tasks such as autonomous driving or improving performance. By using previously demonstrated expert behaviours from advanced drivers, Squadra Corse PoliTo can deploy the trained models to inquire which actions should be taken in new and unseen scenarios. The IL model, `BCNet` works as a "Driver Model", making informed strategy decisions for specific states on track, and can be used for efficient testing of new configurations on the driving simulator and creating personalized feedbacks for drivers to improve their style according to the expert demonstrations.

Chapter 7

Conclusion

The final chapter of the thesis will serve as the final synthesis of the work conducted by discussing the summary of activities, elaboration of results, limitations and future works. The main focus of this project was to develop a holistic strategy optimization framework for Squadra Corse PoliTo, that can extract information about the intrinsic relationships between the driver actions and performance and determine driving styles and the effects of each behavior to build a driver model that can mimic driver's decision-making process. The tools were developed to be deployed by the team to increase performance during the Dynamic Events of a Formula Student event, namely Autocross, Skidpad and most importantly the Endurance round where the driver skill and strategy are crucial.

1. *Outcome Prediction:* Prediction models were built using Regression algorithms to enable testing the outcomes of certain scenarios and driving styles before the actual track day or event. By using these models, important insights were gained about the relationships of input and output variables.

2. *Driving Style Analysis:* Driving styles of SCP drivers were analyzed in terms of Smoothness, Performance, Response and Consistency to understand which set of behaviours performed best. Following these findings, a Driver Identification model was built by using Classification algorithms to underline the findings of DSA, proving the existence of driving styles that were also captured by ML models.

3. *Optimization and Imitation Learning:* Using the expert demonstrations formed by the results of DSA, and substituting a regression formula as an objective function, an optimization model was constructed based on Gaussian Processes and Bayesian Optimization. Following the less than satisfactory results of numerical optimization, Imitation Learning was employed as a remedy for strategy optimization task. Neural Networks were chosen for Behavioral

Cloning task, which is learning by expert demonstrations and imitating the learnt policy on unseen scenarios. A driver model was created that can make informed decisions and the results of this model exceeded expectations set by the Optimization model.

These steps were built incrementally, each following the knowledge extracted in the previous step to construct robust tools for driver modelling, outcome prediction and driver identification. When used in combination, these tools will improve the quality of feedback that can be given to drivers for performance maximization and to present SCP with a deeper understanding of human and machine interactions. If each action made by the driver can be understood in a level that was aimed to be achieved in this thesis, the design, building, testing and driving of the vehicle can be carried out in a significantly more efficient and successful manner.

7.1 Limitations and Future Works

Despite obtaining promising results and extracting insightful knowledge, several limitations were faced throughout this thesis, which present openings for further exploration and improvement in future work. The limitations are mainly a result of lack of time that could be used on this project and lack of research capability associated with limited resources.

Squadra Corse PoliTo is an entity in itself that has deadlines and tight schedule constraints, therefore certain shortcomings on what could realistically be achieved were faced. Datasets lacked diversity in certain aspects such as limited data points for specific scenarios or underrepresented groups. The generalization capability of the ML models may have been damaged due to this reason despite extensive data pre-processing to ensure the integrity.

The time and resource limits constrained the amount of different tools that could be developed given that the thesis subject demands considerable amounts of both of those factors. Another challenge that was faced arising from the limitation mentioned above was the limit on complexity and individual improvement of the models developed. The results potentially could be made better by hyperparameter tuning and higher number of iterations. Due to the time constraint it was not possible to explore alternative methodologies in depth, for example in the Optimization step.

The most crucial and apparent limitation in this thesis is the number of assumptions made to simplify the search space and models. Modeling the complex behavior of human beings, especially in interaction with a system such as a racing vehicle undoubtedly requires several simplifications. These simplifications, while necessary due to constraints in time, computational power, and data availability, may have

impacted the model's ability to successfully map the intricacies of human-vehicle interactions.

The search space of strategy and performance optimization for motorsports domain is vast, with numerous directions that could be chosen to work on. This thesis work has only scratched the surface of possible research that can be realized. Possible directions that can and should be explored further in the future are:

- *Vehicle and Environment Modeling:* Future work could be done to model the vehicle and environment in a more detailed way. This improvement would help replicate the real-world dynamics with higher degrees of freedom.
- *Dataset Augmentation:* Increasing the size and collecting more expert demonstrations representing diverse states and scenarios could increase the variance represented in the dataset, hence increasing the knowledge that can be learned and generalized by the models. By incorporating environmental factors such as changes in weather conditions, various vehicle setup configurations, or different tracks, more comprehensive testing can be made for a variety of driving contexts.
- *Real-World Applications:* By deploying the simulator-data trained models on real-track test data, the robustness and predictive capabilities of the models can be challenged. "Transfer Learning" approach can be used to bridge the gap between simulator and real track data. Output Prediction and Imitation Learning models can be modified to be deployed in real-time to make track or corner-specific predictions. In addition to this, by using interactive visualization techniques during FS events, the driving styles of drivers can be analyzed, and personalized feedback can be given on the spot.
- *Novel Approaches to Driver Modeling:* Future research could improve upon the findings of Imitation Learning by using "GAIL" method in the study of Ho et al. [29]. Another point to consider is the potential adaptation of Reinforcement Learning for driver modeling and strategy optimization. If the dataset, vehicle model and the environmental dynamics are present, Reinforcement Learning can be a strong method for inference.

By addressing these points, future research could improve the current results and further accelerate the ability to optimize Formula Student competition strategy to maximize performance.

List of Tables

List of Figures

Appendix A

Appendix

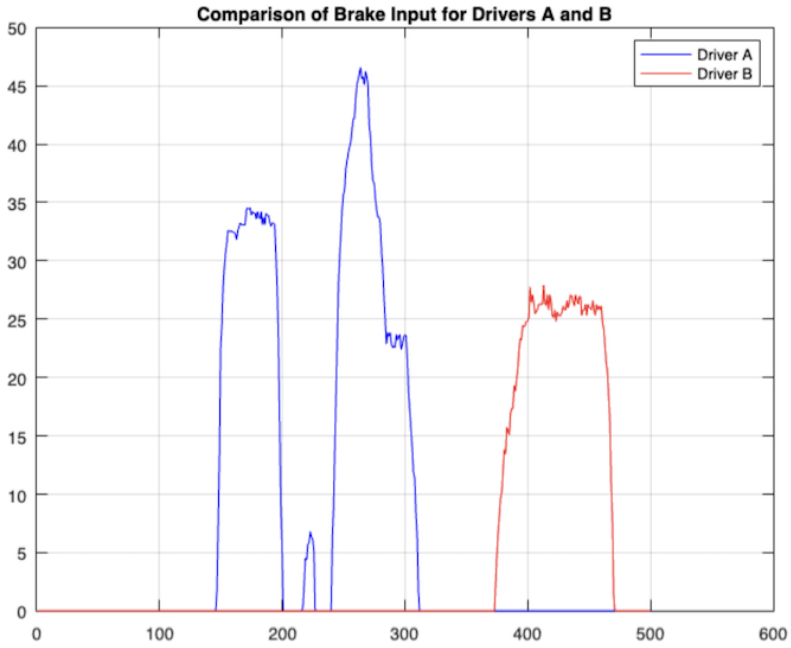

Figure A.1: Brake Signals of Drivers A and B on Corner 2.

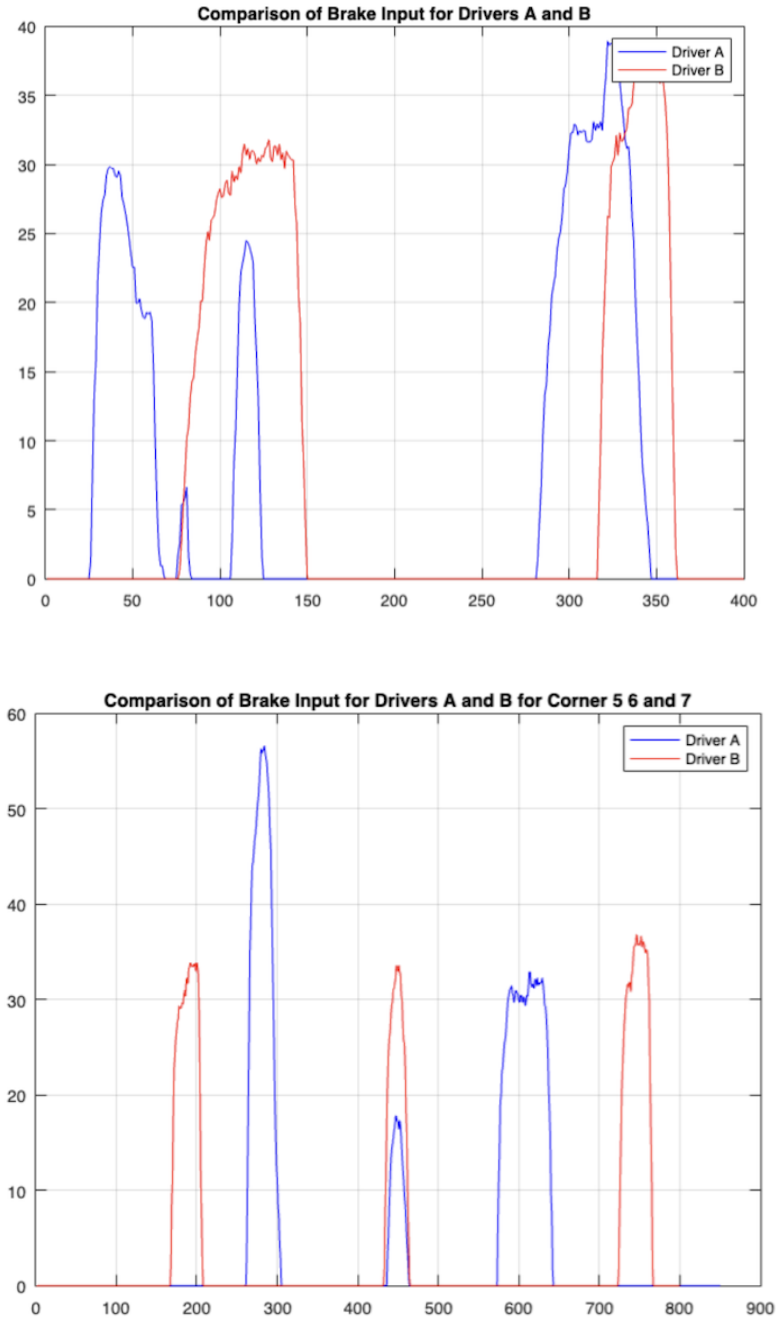

Figure A.2: Brake Signals of Drivers A and B on Corners 4, 5, 6 and 7.

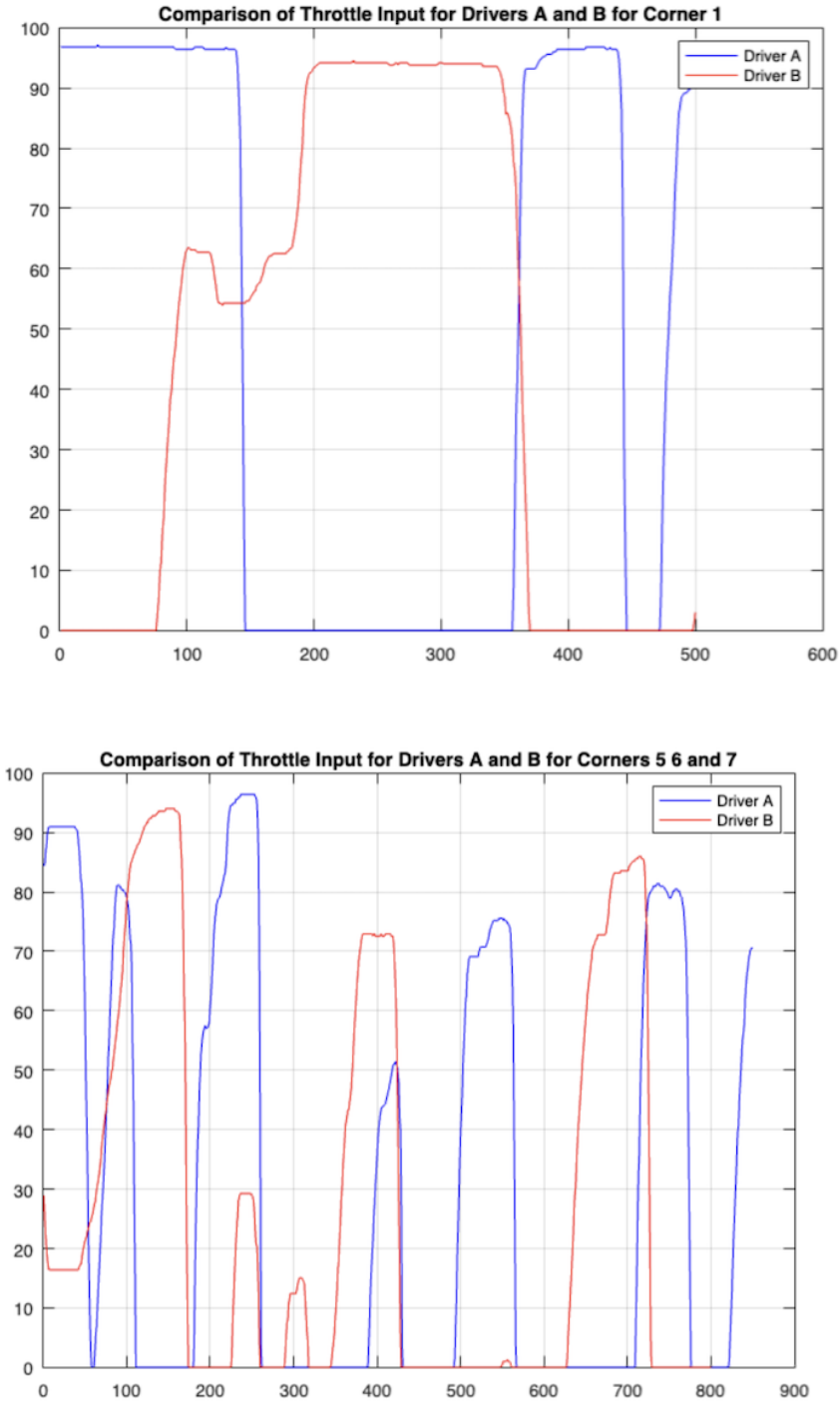

Figure A.3: Throttle Signals of Drivers A and B on Corners 1, 5, 6 and 7.

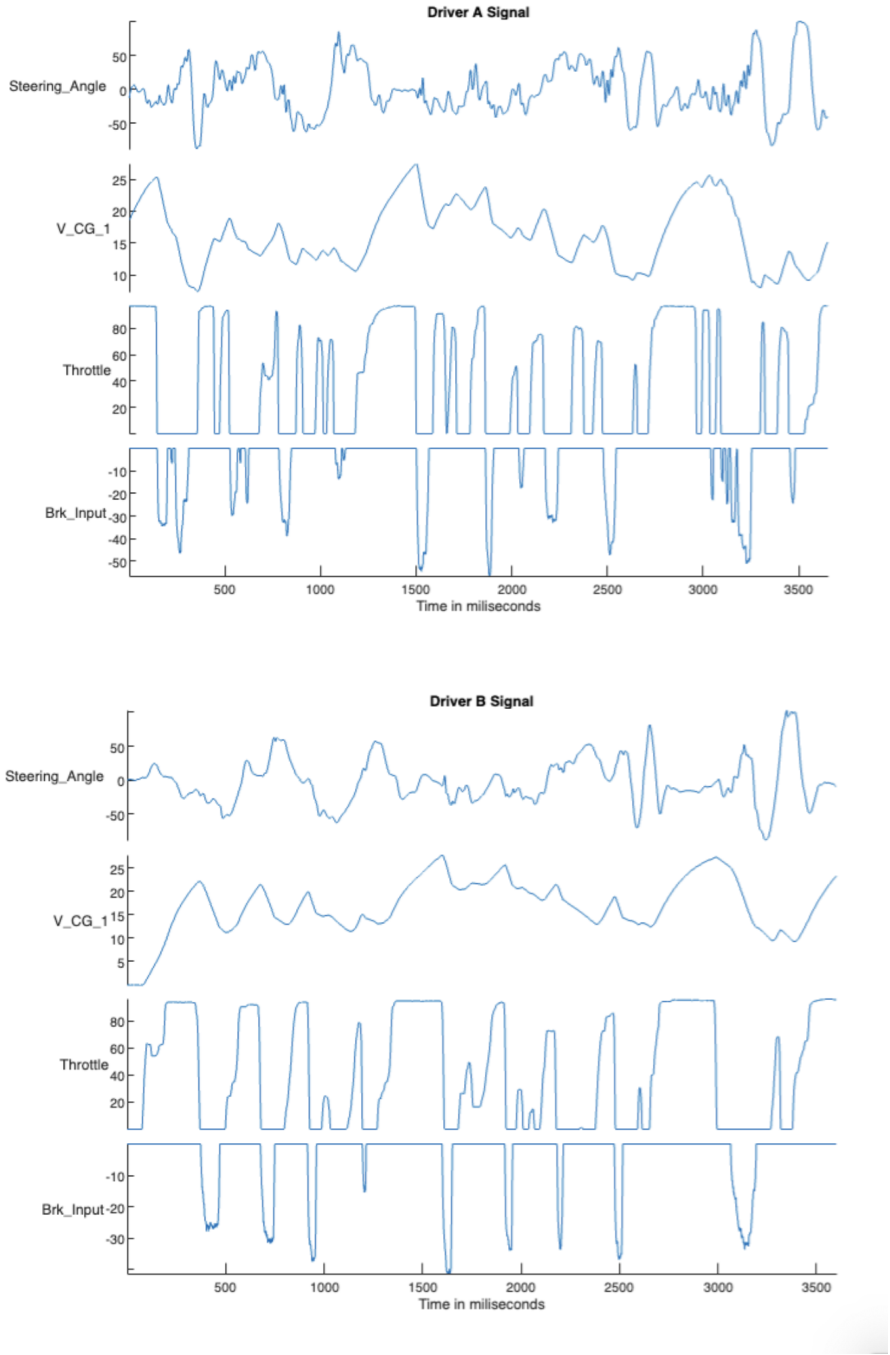

Figure A.4: Feature Signals of Driver A and B for a complete lap.

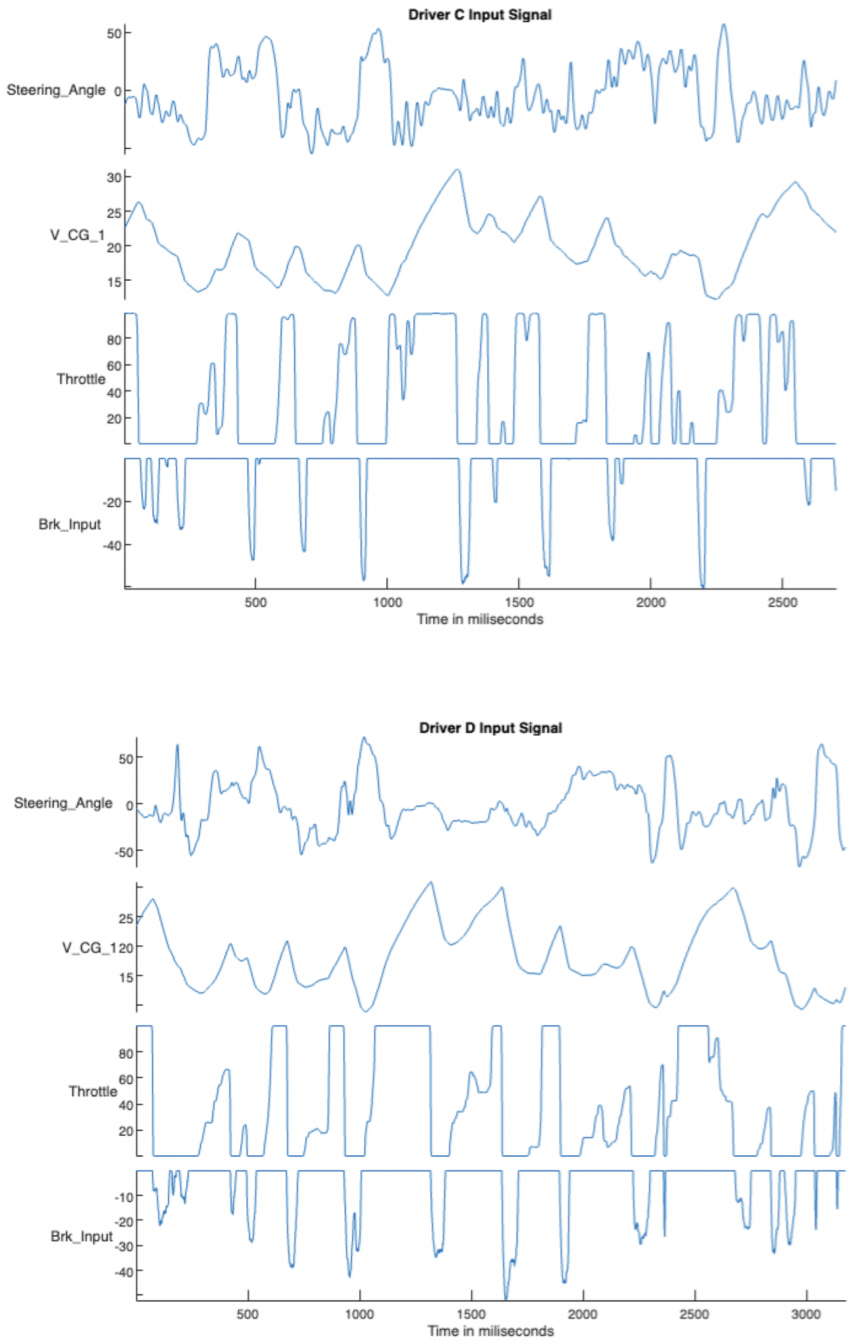

Figure A.5: Feature Signals of Driver C and D for a complete lap.

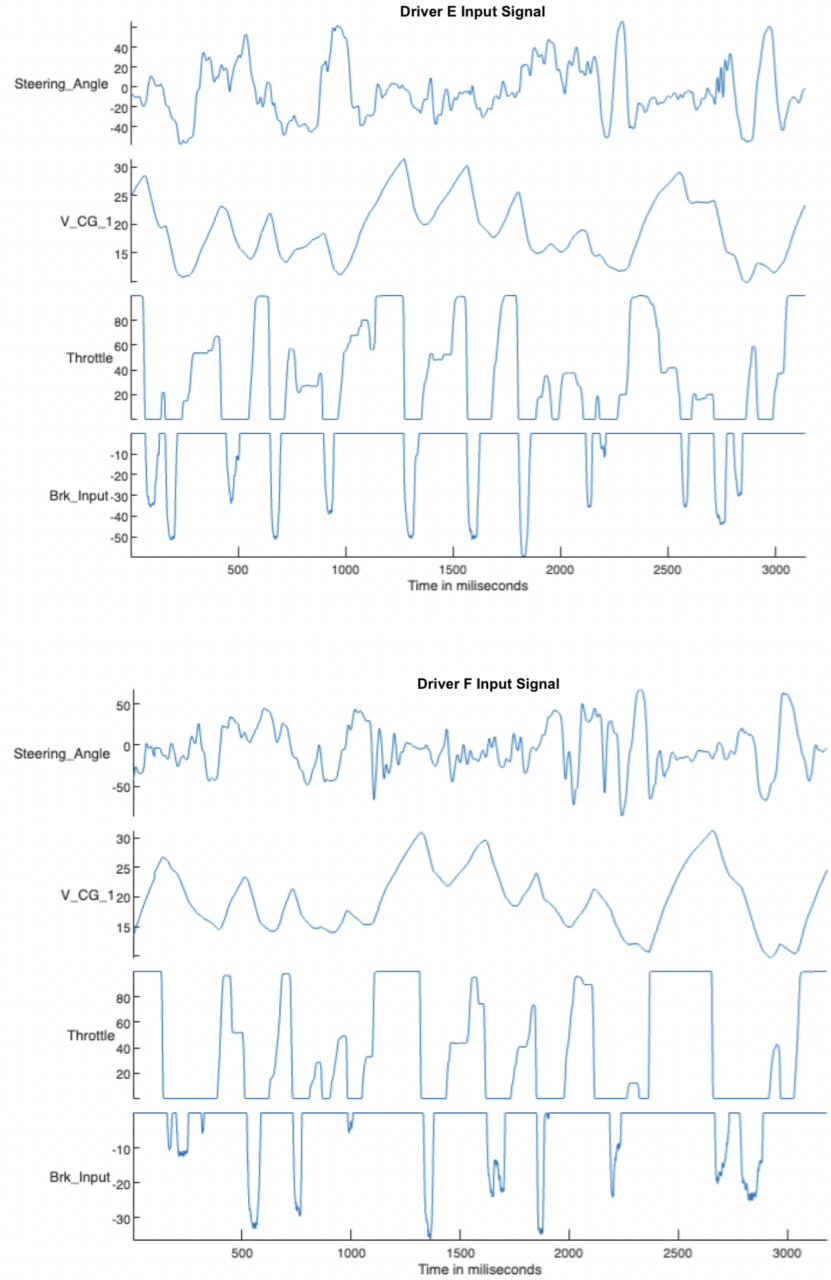

Figure A.6: Feature Signals of Driver E and F for a complete lap.

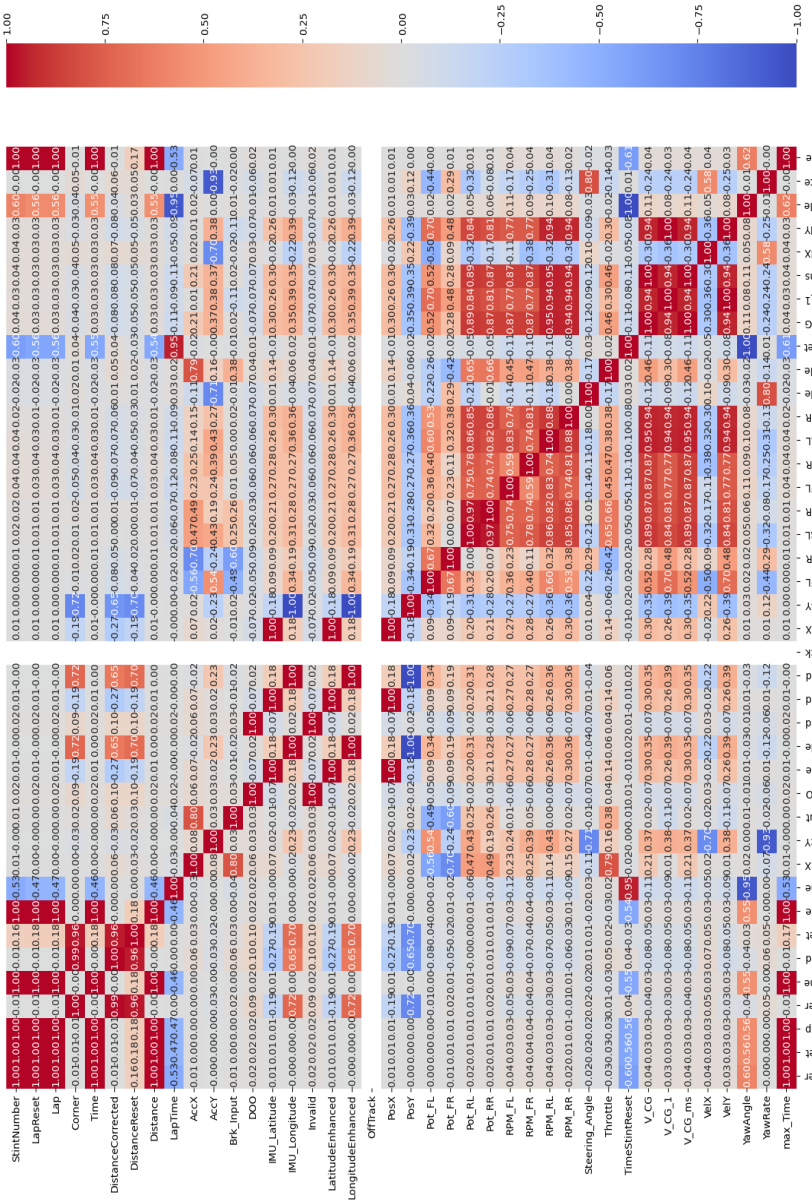

Figure A.7: Correlation Matrix for Simulator Signals Dataset.

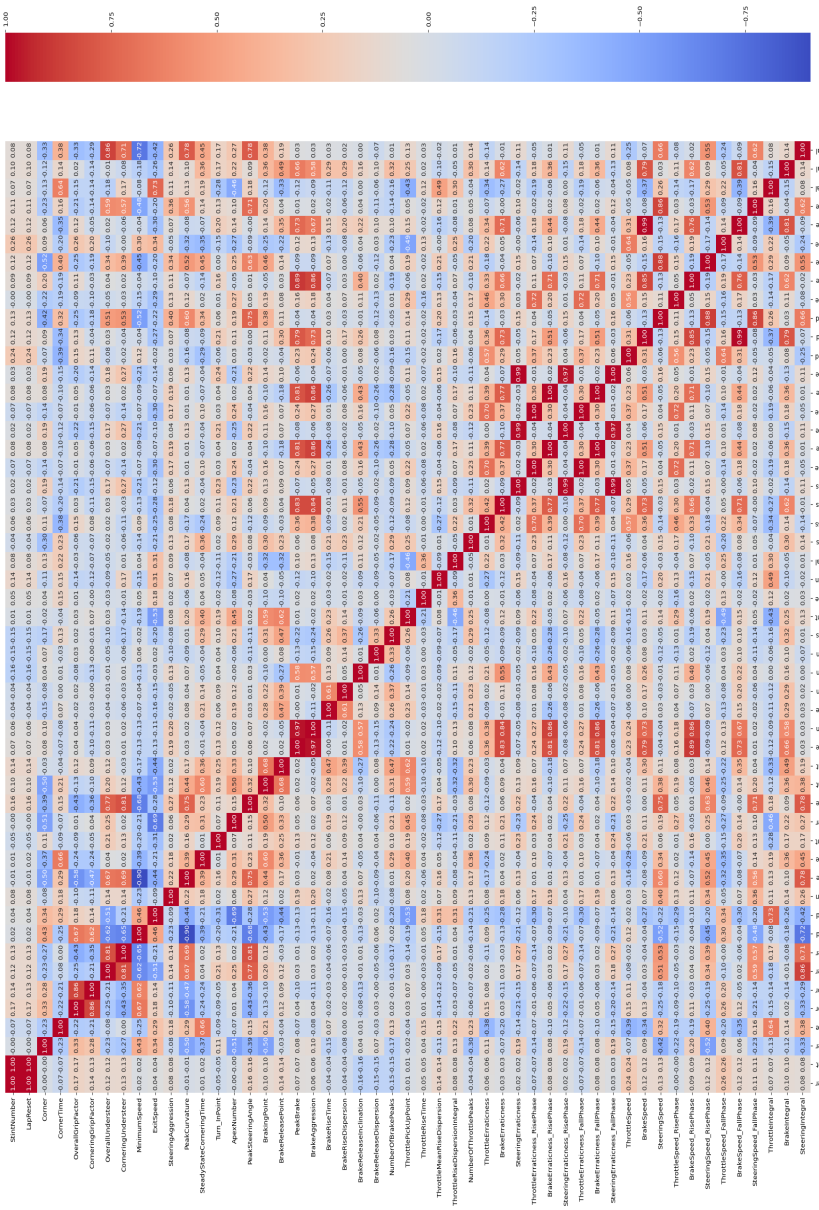

Figure A.8: Correlation Matrix of Corner Indicators Dataset.

Bibliography

[1] *What is Formula Student*. URL: https://formulastudent.ch/what-is-fs.php (cit. on p. 2).

[2] P. Delzell, P. Mccabe, and A. Mourad. *Automation of Data Analysis in Formula 1 Racing* (cit. on p. 10).

[3] D. Rockerbie. «Race to the Podium: Separating and Conjoining the Car and Driver in F1 Racing». In: (). DOI: 10.13140/RG.2.2.24783.61609. URL: https://www.researchgate.net/publication/353692674 (cit. on p. 11).

[4] F. Bi. *How Formula One Teams Are Using Big Data To Get The Inside Edge*. URL: https://www.forbes.com/sites/frankbi/2014/11/13/how-formula-one-teams-are-using-big-data-to-get-the-inside-edge/?sh=582011275588 (cit. on p. 11).

[5] *Every Second Counts: Inside Scuderia Ferrari's AI and Data Tech*. URL: https://www.wired.com/sponsored/story/every-second-counts-inside-scuderia-ferraris-ai-and-data-tech/ (cit. on p. 11).

[6] M. Belrzaeg. «Vehicle dynamics and tire models: An overview». In: *World Journal of Advanced Research and Reviews* 12 (1 Oct. 2021), pp. 331–348. DOI: 10.30574/wjarr.2021.12.1.0524 (cit. on p. 11).

[7] Farroni F., A. Sakhnevych, and F. Timpone. «Physical modelling of tire wear for the analysis of the influence of thermal and frictional effects on vehicle performance». In: *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*. Vol. 231. SAGE Publications Ltd, Feb. 2017, pp. 151–161. DOI: 10.1177/1464420716666107 (cit. on p. 11).

[8] J. Vogel. *Tech Explained: Formula 1 Tyre Model Development*. Sept. 2021. URL: https://www.racecar-engineering.com/articles/tech-explained-formula-1-tyre-model-development/ (cit. on p. 11).

[9] L. Roberts, J. Correia, M. Finnis, and K. Knowles. «Aerodynamic characteristics of a wing-and-flap configuration in ground effect and yaw». In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 230 (6 May 2016), pp. 841–854. ISSN: 20412991. DOI: 10.1177/0954407015596274 (cit. on p. 11).

[10] Kanwar Singh and Srikanth Sivaramakrishnan. «Extended Pacejka Tire Model for Enhanced Vehicle Stability Control». In: (May 2023). DOI: 10.48550/arXiv.2305.18422 (cit. on p. 12).

[11] C. Miyajima, Y. Nishiwaki, K. Ozawa, and T. Wakita. «Driver modeling based on driving behavior and its evaluation in driver identification». In: *Proceedings of the IEEE* 95 (2 2007), pp. 427–437. ISSN: 00189219. DOI: 10.1109/JPROC.2006.888405 (cit. on p. 14).

[12] A. Bell, J. Smith, C. Sabel, and K. Jones. *Formula for success: Multilevel modelling of Formula One Driver and Constructor performance*. June 2016. DOI: 10.1515/jqas-2015-0050 (cit. on p. 14).

[13] C. Macadam. *Understanding and Modeling the Human Driver*. 2003 (cit. on p. 15).

[14] P. Morse. *Driver-in-the-Loop Simulators: Who's the Driver?* URL: https://www.ansiblemotion.com/automotive-driver-in-the-loop-simulation-articles/who-drives-driver-in-the-loop-simulators (cit. on p. 15).

[15] J. Bekker and W. Lotz. «Planning formula One race strategies using discrete-event simulation». In: *Journal of the Operational Research Society* 60 (7 2009), pp. 952–961. ISSN: 14769360. DOI: 10.1057/palgrave.jors.2602626 (cit. on p. 15).

[16] A. Tasora. *REAL-TIME SIMULATION OF A RACING CAR*. URL: https://www.researchgate.net/publication/237297788 (cit. on p. 15).

[17] A. Jung. *Machine Learning: The Basics*. 2023 (cit. on p. 16).

[18] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. 2nd. The MIT Press, 2014 (cit. on p. 16).

[19] Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. «A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges». In: (Sept. 2023). URL: http://arxiv.org/abs/2309.02473 (cit. on p. 16).

[20] *Understanding Support Vector Machine Regression*. URL: https://it.mathworks.com/help/stats/understanding-support-vector-machine-regression.html (cit. on p. 17).

[21] J. Wang. «An Intuitive Tutorial to Gaussian Process Regression». In: (Sept. 2020). DOI: 10.1109/MCSE.2023.3342149. URL: http://arxiv.org/abs/2009.10862%20http://dx.doi.org/10.1109/MCSE.2023.3342149 (cit. on pp. 18, 20).

[22] B. Subhasish. *Introduction to Gaussian Process Regression (GPR)*. 2020. URL: https://subhasish-basak-c-94990.medium.com/introduction-to-gaussian-process-regression-gpr-b6c40f6ef6f9 (cit. on p. 18).

[23] *Gaussian Process Regression Models*. URL: https://it.mathworks.com/help/stats/gaussian-process-regression-models.html (cit. on p. 18).

[24] Andrea Galeazzi, Francesco de Fusco, Kristiano Prifti, Francesco Gallo, Lorenz Biegler, and Flavio Manenti. «Predicting the performance of an industrial furnace using Gaussian process and linear regression: A comparison». In: *Computers and Chemical Engineering* 181 (Feb. 2024). ISSN: 00981354. DOI: 10.1016/j.compchemeng.2023.108513 (cit. on p. 18).

[25] E. Barbierato and A. Gatti. *The Challenges of Machine Learning: A Critical Review*. Jan. 2024. DOI: 10.3390/electronics13020416 (cit. on p. 21).

[26] D. Foster, A. Block, and D. Misra. *Is Behavior Cloning All You Need? Understanding Horizon in Imitation Learning*. 2024. arXiv: 2407.15007 [cs.LG]. URL: https://arxiv.org/abs/2407.15007 (cit. on p. 21).

[27] Z. Lőrincz. *A brief overview of Imitation Learning*. Sept. 2019. URL: https://smartlabai.medium.com/a-brief-overview-of-imitation-learning-8a8a75c44a9c%20%%20%%20Other%20examples (cit. on p. 22).

[28] S. Ross, J. Gordon, and J. Bagnell. *A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning*. 2011. arXiv: 1011.0686 [cs.LG]. URL: https://arxiv.org/abs/1011.0686 (cit. on p. 22).

[29] J. Ho and S. Ermon. *Generative Adversarial Imitation Learning*. 2016. arXiv: 1606.03476 [cs.LG]. URL: https://arxiv.org/abs/1606.03476 (cit. on pp. 22, 90).

[30] P. Frazier. *A Tutorial on Bayesian Optimization*. July 2018 (cit. on p. 23).

[31] J. Brownlee. *How to Perform Feature Selection With Numerical Input Data*. URL: https://machinelearningmastery.com/feature-selection-with-numerical-input-data/ (cit. on p. 23).

[32] S. Turney. *Pearson Correlation Coefficient (r) | Guide and Examples*. Feb. 2024. URL: https://www.scribbr.com/statistics/pearson-correlation-coefficient/#:~:text=The%20Pearson%20correlation%20coefficient%20(r,the%20relationship%20between%20two%20variables.&text=When%20one%20variable%20changes%2C%20the,changes%20in%20the%20same%20direction. (cit. on p. 23).

[33] N. Faizi and Y. Alvi. «Correlation». In: *Biostatistics Manual for Health Research* (Jan. 2023), pp. 109–126. DOI: 10.1016/B978-0-443-18550-2.00002-5. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780443185502000025 (cit. on p. 24).

[34] S. Swarup. *Anomaly Detection with GESD (Generalized Extreme Studentized Deviate) in Python*. Apr. 2021. URL: https://towardsdatascience.com/anomaly-detection-with-generalized-extreme-studentized-deviate-in-python-f350075900e2 (cit. on p. 24).

[35] C. Richter, M. O'Reilly, and E. Delahunt. *Machine learning in sports science: challenges and opportunities*. 2021. DOI: 10.1080/14763141.2021.1910334 (cit. on p. 26).

[36] T. Tulabandhula and C. Rudin. *TIRE CHANGES, FRESH AIR, AND YELLOW FLAGS: CHALLENGES IN PREDICTIVE ANALYTICS FOR PROFESSIONAL RACING*. June 2014. DOI: 10.1089/big.2014.0018 (cit. on pp. 26, 27).

[37] R. Bunker and F. Thabtah. «A machine learning framework for sport result prediction». In: *Applied Computing and Informatics* 15 (1 Jan. 2019). Fig.4 is useful as a path., pp. 27–33. ISSN: 22108327. DOI: 10.1016/j.aci.2017.09.005 (cit. on p. 26).

[38] E. Stoppels. *Predicting Race Results using Artificial Neural Networks*. 2017 (cit. on p. 26).

[39] L. Tejada. *Applying Machine Learning to Forecast Formula 1 Race Outcomes*. 2019 (cit. on p. 26).

[40] H. Sicoie. *Machine Learning framework for Formula 1 race winner and championship standings predictor* (cit. on pp. 26, 27).

[41] V. Kumar and G. Jacek. *Formula 1 Race Analysis using Machine Learning* (cit. on p. 27).

[42] B. Pfitzner. *Do Reliable Predictors Exist for the Outcomes of NASCAR Races?* 2008. URL: http://jayski.thatsracin.com/ (cit. on p. 27).

[43] T. Rishel and B. Baker E. Pfitzner. *FINISHING OR WINNING? THE VARIABLES THAT IMPACTED THE NASCAR CHAMPIONSHIP IN THE CHASE I FORMAT (2004-2013)*. 2004 (cit. on p. 27).

[44] J. Schleinitz, T. Schwarzhuber, and L. Wörle. «Race Driver Evaluation at a Driving Simulator using a physical Model and a Machine Learning Approach». In: (Jan. 2022). URL: http://arxiv.org/abs/2201.12939 (cit. on p. 28).

[45] J. Segers. *Analysis Techniques for Racecar Data Acquisition Second Edition*. 2014. DOI: 10.4271/R-408. URL: http://books.sae.org. (cit. on pp. 28, 36, 43, 62, 70).

[46] F. Hojaji, A. Toth, and M. Campbell. «A Machine Learning Approach for Modeling and Analyzing of Driver Performance in Simulated Racing». In: *Communications in Computer and Information Science*. Vol. 1662 CCIS. Springer Science and Business Media Deutschland GmbH, 2023, pp. 95–105. ISBN: 9783031264375. DOI: 10.1007/978-3-031-26438-2_8 (cit. on p. 29).

[47] S. Löckel, J. Peters, and P. Vliet. «A Probabilistic Framework for Imitating Human Race Driver Behavior». In: (Jan. 2020). DOI: 10.1109/LRA.2020.2970620. URL: http://arxiv.org/abs/2001.08255%20http://dx.doi.org/10.1109/LRA.2020.2970620 (cit. on pp. 29, 30).

[48] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni. «Race driver model». In: *Computers and Structures* 86 (13-14 July 2008), pp. 1503–1516. ISSN: 00457949. DOI: 10.1016/j.compstruc.2007.04.028 (cit. on p. 30).

[49] Wei, Hongchuan, Weston Ross, Stefano Varisco, Philippe Krief, and Silvia Ferrari. «Modeling of human driver behavior via receding horizon and artificial neural network controllers». In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 6778–6785. DOI: 10.1109/CDC.2013.6760963 (cit. on p. 30).

[50] O. Benderius. *Driver modeling: Data collection, model analysis, and optimization* (cit. on p. 30).

[51] M. Boettinger and D. Klotz. «Mastering Nordschleife – A comprehensive race simulation for AI strategy decision-making in motorsports». In: (June 2023). URL: http://arxiv.org/abs/2306.16088 (cit. on p. 31).

[52] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern. «Formula RL: Deep Reinforcement Learning for Autonomous Racing using Telemetry Data». In: (Apr. 2021). URL: http://arxiv.org/abs/2104.11106 (cit. on p. 31).

[53] S. Ju, P. Vliet, O. Arenz, and J. Peters. «Digital Twin of a Driver-in-the-Loop Race Car Simulation with Contextual Reinforcement Learning». In: *IEEE Robotics and Automation Letters* 8 (7 July 2023), pp. 4107–4114. ISSN: 23773766. DOI: 10.1109/LRA.2023.3279618 (cit. on p. 31).

[54] A. Heilmeier, A. Thomaser, M. Graf, and J. Betz. «Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport». In: *Applied Sciences (Switzerland)* 10 (21 Nov. 2020), pp. 1–32. ISSN: 20763417. DOI: 10.3390/app10217805 (cit. on p. 31).

[55] S. Jimenez and A. Luchsinger. *Towards Formula One Driver/Vehicle Optimization* (cit. on p. 31).

[56] S. Pontin. *AI-Based Race Strategy Assistant and Car data Monitor*. 2023 (cit. on p. 31).

[57] X. Liu, A. Fotouhi, and D. J. Auger. «Formula-E race strategy development using distributed policy gradient reinforcement learning». In: *Knowledge-Based Systems* 216 (Mar. 2021). ISSN: 09507051. DOI: 10.1016/j.knosys.2021.106781 (cit. on p. 31).

[58] A. Tatulea-Codrean, T. Mariani, and S. Engell. «Design and simulation of a machine-learning and model predictive control approach to autonomous race driving for the F1/10 platform». In: *IFAC-PapersOnLine*. Vol. 53. Elsevier B.V., 2020, pp. 6031–6036. DOI: 10.1016/j.ifacol.2020.12.1669 (cit. on p. 31).

[59] D. Piccinotti. *Open Loop Planning for Formula 1 Race Strategy identification*. 2020 (cit. on p. 32).

[60] D. Piccinotti, A. Likmeta, N. Brunello, and M. Restelli. *Online Planning for F1 Race Strategy Identification*. 2021. URL: www.aaai.org (cit. on p. 32).