



PRIFYSGOL CYMRU

Y Drindod Dewi Sant

UNIVERSITY OF WALES

Trinity Saint David

Enhancing Motorcycle Racing Performance through Telemetry Data Predictive Modeling: Machine Learning and Chatbot-Assisted Optimization

Iriney Svilen Georgiev

1900719

Master's Project

ACFA7003

Swansea

September 2023

Course

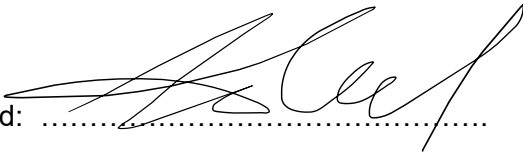
MSc Motorcycle Engineering

Supervisor

Dr Owen Williams

Author's Declaration

This thesis is the sole work of the author and is the result of original research. All works completed have been done so by the author, using resources from University of Wales Trinity Saint David and/or Project Partner Organisations. It has not been previously submitted for examination which has led to the award of a degree.

Signed: 

Date: 

Abstract

This study examined 695 laps at the Phillip Island circuit on a ZX6R Kawasaki motorcycle designed for the SuperSport class of the FIM Superbike World Championship (WSBK). The dataset was obtained from MoTeC i2 Pro telemetry software and rigorously preprocessed with a Python 3.11 algorithm. As part of this process, missing values were addressed, categorical variables were encoded and numerical attributes were standardized. In this way, the dataset was prepared for effective use with the Random Forest Regressor, a machine learning model (ML) based on artificial intelligence (AI) (Scikit-learn, 2023). Additionally, feature engineering was applied to indicate session IDs, lap counts and sector divisions, enhancing both the model's precision and computational efficiency.

The ML model, when tested with unseen data, successfully identified the best lap and generated an ideal lap utilizing the best sectors to identify the optimal lap time performance. Furthermore, the model managed to formulate optimal tracing solutions for each channel with which it was trained so as to beat the lap time record for the circuit. To facilitate a deeper understanding, OpenAI's GPT-4 chatbot was integrated, which converted complex data points into comprehensible narratives (OpenAI, 2023). The findings were showcased via a user-friendly interface (Plotly, 2023). Validation of the final product was accomplished through meticulous cross-referencing with test dataset setting sheets and with the assistance of a data engineer who has over two decades of domain expertise.

Approximately 70% of the model's forecasts were in accordance with the engineer's assessment. A significant portion of the variance was attributed to the model's focus on primary channels such as GPS speed, engine speed, throttle position and others. Such focus indicated that potential enhancement areas might have been missed, warranting further investigation and emphasis in the future. Adding training data, backed by robust computing resources, may improve model's performance in a limitless way. Nevertheless, the telemetry ML prediction model was able to analyze hundreds of laps within seconds. In addition, the AI chatbot provided critical insights into areas for potential performance enhancement. Thus, the final product has established itself as an indispensable tool for motorcycle racing teams to interpret telemetry data.

Acknowledgements

I would like to express my sincere gratitude to the individuals who have been instrumental in the realization of this project:

Dr. Owen Williams, my esteemed supervisor, whose unwavering support, expert guidance, and profound insights into the motorcycle racing world have proven to be invaluable.

Andrew Harrison, a lecturer at UWTSD, who not only generously provided the telemetry data that was crucial to my work but also offered his vast expertise in data analysis to validate my findings.

The IV Racing team, whose expertise was critical to the development of the telemetry prediction machine learning model.

My parents, Gergana Georgieva, Svilen Georgiev and my brother Teodosii Georgiev, who have given me all their belief and support to pursue my dreams.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	2
1.2.1	Data Collection and Preprocessing	2
1.2.2	Model Development and Evaluation	3
1.3	Aims and Objectives	4
1.4	Overview	4
2	Literature Review	6
2.1	Foundation	6
2.2	Natural Language Processing	9
2.3	Telemetry Analysis Modelling	10
2.4	Performance Optimization	13
3	Methodology	17
3.1	Data Collection	17
3.2	Data Preprocessing	18
3.3	Feature Engineering	21
3.4	Development	23
3.4.1	Lap Time Prediction Model	23
3.4.2	Telemetry Prediction Model	25
3.4.3	Chatbot	32
4	Results	37
4.1	Fast Against Ideal Lap	37
4.2	Ideal Against Predicted Lap	39
5	Discussion	41
5.1	Rider Performance Data Analysis	41
5.2	Further Improvement Data Analysis	42

6	Conclusions	44
6.1	Findings	44
6.2	Future Work	45
6.3	Reflection.....	46
	References	48
7	Appendices	52
7.1	Appendix A	52
7.2	Appendix B	52
7.3	Appendix C	53
7.4	Appendix D	54
7.5	Appendix E	56
7.6	Appendix F	58

1 Introduction

The term machine learning (ML) refers to a subset of artificial intelligence (AI) in which algorithms are trained to recognize patterns in data, allowing them to make predictions or decisions without being explicitly programmed. As a result of the data they process, computers acquire the ability to improve their performance on a task over time. The application of ML has gained prominence across a broad range of industries, including finance, healthcare and most recently motorsport (Massegur, 2022).

Telemetry data provides insight into the intricate world of motorcycle dynamics, enabling data engineers to refine mechanical configurations and riding techniques to improve lap times (Segers, 2014). This master's project aims to develop telemetry analysis software that combines advanced ML methodologies with the nuanced finesse of natural language processing. Consequently, assist data engineers in analyzing telemetry data almost instantly. With this capability, resources can be strategically allocated to the most critical areas of optimization, resulting in faster lap times.

1.1 Motivation

Previously, Georgiev (2022) developed a design of experiment (DOE) model that integrated a chassis simulation with a two-dimensional (2D) quasi-steady-state (QSS) model of a riderless motorcycle. This software package was designed to guide motorcycle engineers in deriving baseline setups for specific circuits and identifying pivotal setup modifications. A notable issue arose during the validation phase, the chassis configuration did not always resonate with the rider's preferences. Around 80% of the time, there was a misalignment that adversely affected their riding techniques and racing trajectory. This deviation occurred due to the limitations of the QSS model, which failed to consider aspects such as surface elevation, rider's body influence and race line flexibility.

In preparation for this master's project, a comprehensive literature review of motorcycle modeling techniques to enhance the foundational framework (Georgiev, 2023). A significant advance in the field was presented by Biniewicz and Pyrz (2023), who introduced QSS motion equations and real-world data-driven input constraints. Their approach relied on differential equations to distinguish positive and negative longitudinal accelerations, as well as statistically derived jerk constraints that considered vehicle dynamics, mass transfer, moment of inertia and driver

capabilities. However, the model proved to be unreliable and labor-intensive despite its versatility. This was due to the continuous loop of refinement and intricate details required for each chassis setup, as well as the influence of track and weather conditions.

Initially, this master's project aimed to integrate ML with Biniewicz and Pyrz's (2023) approach. By employing ML techniques, the laborious refinement processes could be dramatically streamlined, paving the way for profound, scalable insights. Yet, extensive training data was essential to the development of an effective ML model. As Aggarwal (2018) suggests, the dataset should exceed the model parameters by at least ten times, meaning thousands of laps might be essential for a solid representation. Securing even a foundational dataset, such as 500 laps, presents challenges given the fluctuations in bike configuration, rider proficiency and meteorological conditions, prompting a strategic reassessment of the project's direction.

While traditional research has focused on replicating "optimal" conditions over a full array of laps, the ML model used in this study focused more on understanding the technical characteristics of the bike and circuit. Concurrently, the model's performance was enhanced when exposed to a diverse dataset encompassing a variety of variables. Likewise, as race riders internalize circuit layouts and become familiar with their machines, an iterative experimentation process follows, involving a myriad of mechanical and riding configurations, culminating in the pursuit of pinnacle performance. The model's predictive capability was based on a more holistic and adaptable framework, based on both established literature and the previous work of Georgiev (2023).

1.2 Problem Statement

The analysis of motorcycle telemetry data using predictive ML algorithms involves a number of layers. Each phase, from meticulous data acquisition to nuanced rider performance interpretation, poses its own set of challenges. While some complexities were anticipated at the project's onset and are discussed in this section, others emerged as the research progressed.

1.2.1 Data Collection and Preprocessing

Data collection is not merely about collecting vast numbers of datasets, but also about ensuring that the data is collected in a uniform manner and under consistent conditions (Aggarwal, 2018).

For the purpose of the project, telemetry data should be obtained from identical tracks using the same motorcycle. This consistency is crucial to the accuracy and adaptability of the model across a variety of scenarios. The ever-changing dynamics of racing environments, coupled with the confidentiality practices of racing teams, make it difficult to achieve this level of uniformity.

Depending on the frequency of data transmission, a single session encompassing 10 laps can generate millions of data points for each sensor. As opposed to conventional binary data, which operates on a simple "yes" or "no" response, this dataset requires the model to discern intricate relationships between recorded channels and lap times to identify "good" or "bad" behaviors. However, the effectiveness of ML models can be compromised if they mistakenly interpret noise or minor fluctuations as important data patterns. Preprocessing techniques can address these challenges, but there must be a delicate equilibrium. Excessive refinement might strip away nuanced data, while insufficient preprocessing might leave inaccurate outliers (Scikit-learn, 2023).

1.2.2 Model Development and Evaluation

Data science and machine learning involve experimenting with a variety of methodologies before settling on the most effective one. Theoretically sound approaches may not yield the desired results when applied to the particular dataset in question. As an example, while certain features may appear crucial, their relevance may be minimal with respect to the data, which may result in their exclusion from subsequent models. Moreover, a shift from one modeling technique to another is driven by the desire to obtain greater accuracy, interpretability and insight (Hastie, et al., 2001).

Having selected the optimal ML model and completed its training, it is crucial to adjust its parameters to achieve the best performance on unfamiliar data. This step is known as hyperparameter tuning. The testing procedure is, however, computationally intensive and thorough, presenting a challenge considering the project's resources and timeline (Bergstra & Bengio, 2012).

Given the novelty of the concept, evaluating the model's performance presents a unique challenge. Traditional metrics of evaluation, such as accuracy, may not adequately demonstrate the model's real-world effectiveness. The testing dataset should ideally be paired with corresponding setting sheets. These documents not only provide information about mechanical setups, but also outline the problems faced and solutions employed during a race session. Due

to team confidentiality concerns, some data may lack these crucial setting sheets. The absence of these variables in even a subset of the testing data might obscure a full analysis of the model, potentially resulting in an oversight of its true capabilities.

1.3 Aims and Objectives

A meticulously crafted Gantt chart, ensuring systematic progression and risk mitigation, outlines the primary milestones of this project and can be found in Section 6.3. In addition to this structured framework, the research unfolds its aims through the defined objectives:

- Collect robust telemetry data, independent of the acquisition system employed. Prioritize datasets obtained from the same circuit and motorcycle. Determine the channels most frequently utilized for data analysis, ensuring their presence across all datasets.
- Identify the appropriate format for converting the telemetry data to be compatible with the ML model. Remove anomalies such as in-and-out laps, crashes, and other potential disruptions that could compromise the model's performance.
- Determine an appropriate ML model and develop an efficient architecture to identify the fastest lap from unobserved data. Assess the influence of various channels on lap time. Based on the rider's objectives, forecast the ideal trace for each channel. Then compare the rider's real-time performance to these standards.
- Select a user-centric chatbot interface and integrate it with the ML algorithm to identify the key discrepancies in channel traces. Fine-tune the chatbot to offer context-specific recommendations for either mechanical adjustments or riding strategies, while using accurate terminology.
- Benchmark the ML model's predictive precision and efficiency against existing methodologies. Analyze its effectiveness by cross-referencing its results with testing data supported by race sheets. Showcase the advantages of this innovative approach in terms of user engagement and procedural flexibility, as well as future directions for research.

1.4 Overview

Chapter 2 focuses on the application of ML and NL models to automate the processing and analysis of data. Several industries are examined in order to highlight the primary methodologies and challenges associated with these technologies.

Chapter 3 delves into the complexities of developing a telemetry predictive ML model, detailing the challenges faced and solutions applied. It also discusses the incorporation of different OpenAI's chatbots and the prompt design for precise application interpretation.

Chapter 4 showcases the results of the ML model's visual telemetry predictions in comparison to the rider's ideal lap to surpass the Phillip Island circuit record. Furthermore, the chatbot's interpretations of these outcomes are validated against the session's setting sheets and assessed by an expert data engineer with 20 years of motorcycle racing experience.

In summary, this work makes the following contributions:

- Determines the quickest lap from a telemetry dataset.
- Generates an optimal lap trace based on the rider's best sectors for each channel that it has been trained on.
- Recreates traces for every analyzed channel based on specific goals.
- Provides an overall performance evaluation based on the rider's best lap and the optimal lap.
- Identifies areas that require further attention to achieve faster lap times by comparing the predicted with the ideal channel traces.
- Introduces a ML model with an OpenAI chatbot interface adaptable to any circuit or motorcycle type. This model swiftly analyzes new data, aiding data engineers in pinpointing performance enhancement areas.

2 Literature Review

Automation of data processing and analysis, celebrated for its precision, efficiency and consistency, has catalyzed transformations across a wide range of industries (Sarker, 2021). However, in motorcycle racing, the terrain is less explored, and the challenge lies in the relatively recent adoption of automated telemetry data interpretation. Literature pivotal to the project is reviewed in this section, emphasizing each work's relevance and depth based on its significance. Thus, a framework and model have been developed, and benchmarks have been established for evaluating the project's results.

2.1 Foundation

Navigating the intricate landscape of data preprocessing and processing is crucial to ensuring predictive ML models' accuracy and reliability. The study by Siploa (2017) provides a pioneering and illuminating insight into electricity consumption forecasting for household appliances utilizing advanced data processing and neural networks. A vital component of this research was the Residential Electrical Findings and Intelligent Tweaking (REFIT) dataset, which collected data on high-frequency electricity consumption from 20 UK households. The research utilized Python 2.7, supplemented by specialized libraries. Specifically, NumPy excelled at facilitating mathematical and numerical operations, streamlining scientific computations, whereas Pandas brought sophisticated data structures and analytical methods tailored for in-depth data analysis. A further advancement to the study's capabilities was the integration of Keras, a user-friendly platform that facilitates the creation and training of neural networks. The combination of different architectures and hyperparameters allowed for streamlined experimentation.

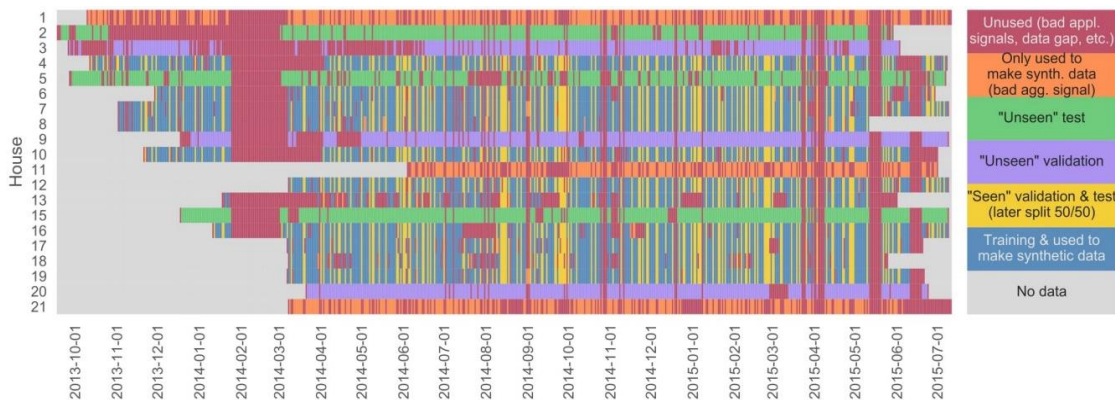


Figure 1: Split between training, validation and test datasets (reproduced from Sipola, 2017).

To perform the preprocessing step of the data modeling, Scikit-learn was utilized and a novel sliding window technique was introduced, as shown in Figure 1. The input data and the corresponding targets were grouped into training, validation, and test subsets. This segmentation served to enhance model stability and counteract overfitting tendencies. To identify the most suitable network architecture and pinpoint hyperparameters, the research combined a dynamic architecture generation technique with both grid search optimization and manual experimentation. Due to the depth and complexity of the NN models, a substantial amount of computational power was required. Relying solely on the REFIT dataset, concerns were also raised regarding the model's ability to be adapted to a wide range of datasets (Sipola, 2017).

In 2017, Nguyen undertook an innovative study utilizing telematics data, which encompasses details on vehicle behaviors like acceleration, braking, and speed sourced from specialized sensors and communication tools. To achieve an accuracy rate of 90% for vehicle recognition, the researcher created a model in Python using popular ML libraries, such as Scikit-learn and pandas. An extensive dataset providing information from over 100,000 vehicle trips was central to this work, which was divided into 80%-20% training and testing.

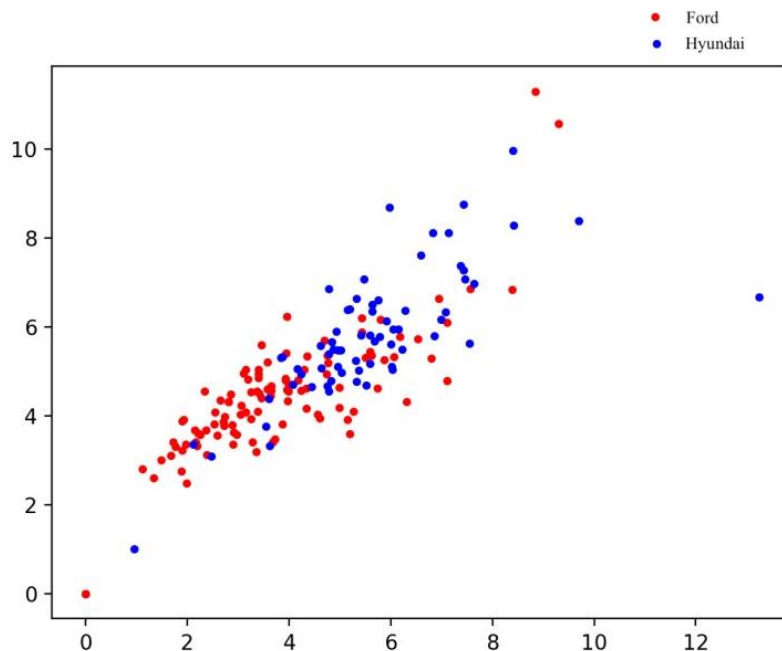


Figure 2: Power-to-Weight ratio with the horizontal axis representing mean and vertical axis indicating deviation (reproduced by Nguyen, 2017).

The vehicle model was predicted using a multiclass Random Forest (RF) classifier based on features from the telematics dataset. The classifier operated on supervised learning, which

involved training the model on a labeled dataset to make predictions. A key aspect of this model was the Power-to-Weight ratio method, which is shown in Figure 2. The ratio, which was calculated for each valid sample, allowed the distinction of vehicle types based on their acceleration and engine responsiveness. However, the study encountered several obstacles, including potential misclassification of vehicles with similar characteristics and scalability problems because of noisy data. To enhance the integrity of this work, the author suggested a deeper dive into ML methodologies and a more intensive validation across a variety of vehicle types and scenarios.

Aside from the exciting advancements in ML methodology, the research conducted by Nguyen (2017) with telematics data only sets the stage for further investigations of optimizing model precision. Versloot (2019) explored the domain of buried utility characterization by utilizing the most common non-linear ML predictive models. Employing the GprMax software, synthetic Ground Penetrating Radar (GPR) data was generated, underpinned by Python, and bolstered by prominent open-source libraries. This led to the crafting of a synthetic dataset, encompassing 1,000 samples with each containing 1,024 data points. To ensure the precision of the model, the training, validation and testing datasets were properly segmented. Afterwards, the research evaluated three state-of-the-art ML models: Random Forest (RF), Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP).

$$\text{Learning Rate Decay} = \frac{\text{Default Learning Rate}}{\text{Number of Epochs}} \quad \textbf{Equation 1: Linear decay rate (Versloot, 2019).}$$

This study was unique in that it applied the learning rate range test during the pre-training phase. With this technique, encapsulated in Equation 1, the learning rate can be automatically adjusted throughout training. By dividing the initial "Default Learning Rate" by the number of epochs, the rate decreases progressively. In the context of machine learning, an epoch refers to the completion of a complete cycle of the training dataset through the learning algorithm. In accordance with this formula, learning rates decrease as training progresses, thereby facilitating rapid convergence with gradual refinement of model parameters. This innovative methodology significantly enhanced the depth of the study, yielding compelling results. In particular, the MLP model achieved an average accuracy of 80%, followed by the RF (69.09%) and SVM (63.64%) models. The MLP's standout performance was linked to its adeptness at decoding complex correlations between input and output parameters. However, the researchers noted variances in model efficacy depending on the utility and material in focus. A suggestion was made to further

evaluate the ML models using real-world data, since synthetic datasets are generally used to demonstrate the potential of a model rather than its full range of applications (Versloot, 2019).

2.2 Natural Language Processing

The 2020 study by Brown et al. deviated from conventional ML research. Their primary focus was on Generative Pre-trained Transformer 3 (GPT-3), a model that does not predict or model relationships between independent and dependent variables in the traditional sense. Rather, the essence of GPT-3 lies in its ability to understand and generate text that is similar to that of a human. This characteristic is a result of its extensive training in linguistic patterns, which allows it to mimic the nuanced articulations of human languages.

The purpose of their investigation was to assess the performance of GPT-3 in few-shot learning scenarios. This scenario involves a ML model performing tasks with a minimal set of examples. When a single example is provided, it is termed one-shot learning, and when the model is based solely on natural language instruction without any prior examples, it is termed zero-shot learning. To demonstrate GPT-3's adaptability, the model was subjected to translation tasks encompassing English to French, German and Romanian. The model's training data was primarily English (93% by word count), and only 7% of the text was in other languages (Brown, et al., 2020).

Table 1: GPT-3 results compared with unsupervised and supervised language predictive models (reproduced by Brown, et al.,2020).

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	45.6^a	35.0 ^b	41.2^c	40.2 ^d	38.5^e	39.9^e
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ ⁺ 19]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG ⁺ 20]	-	-	<u>29.8</u>	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

Table 1 illustrates that GPT-3 demonstrated commendable performance in few-shot learning situations, particularly when compared with standards set by traditional fine-tuning approaches. As a result, the few-shot learning approach was deemed more versatile and resource-efficient

than traditional fine-tuning, which often requires large and specialized datasets for each distinct task (Brown, et al., 2020).

While Brown et al. (2020) laid the foundation for understanding the efficacy of few-shot learning within the GPT-3 model, a study conducted by Schick and Schutze (2021) added an extra layer by introducing an innovative technique aimed at enhancing the performance of ML models within Natural Language Processing (NLP) tasks. A technique termed "hot learning" was employed, where several language learning models were trained with prompts that provided context and guidance. For evaluation, data was sourced from 30 non-native English speakers who interacted with conversational agents over a 30-minute period. Metrics such as the number of correct responses, response times, and user satisfaction levels were meticulously recorded.

Table 3: Performance comparison of few-shot learning models supported with automatically generated prompts (reproduced by Schick and Schutze, 2021).

Method	ADE	B77	NIS	OSE	Over	SOT	SRI	TAI	ToS	TEH	TC	Avg
GPT-2	60.0	12.1	56.1	24.5	49.8	38.0	49.2	61.2	49.8	31.1	72.3	45.8
GPT-Neo	45.2	14.9	40.8	34.3	68.1	40.6	49.3	60.5	56.5	55.4	63.6	48.1
AdaBoost	54.3	02.3	62.6	47.5	83.8	45.5	50.6	55.6	56.0	44.3	62.5	51.4
snlt	60.3	24.8	58.5	30.2	83.1	33.6	49.2	62.6	54.0	44.9	79.1	52.8
GPT-3	68.6	29.9	67.9	43.1	93.7	76.9	51.6	65.6	57.4	52.6	82.1	62.7
SetFit	72.6	53.8	87.2	52.1	90.7	68.2	49.3	62.8	62.0	53.2	83.7	66.9
PET	82.2	59.3	85.7	64.6	90.8	81.6	49.3	63.8	57.6	48.3	82.4	69.6
Human	<u>83.0</u>	<u>60.7</u>	85.7	<u>64.6</u>	91.7	<u>90.8</u>	46.8	60.9	<u>62.7</u>	<u>72.2</u>	<u>89.7</u>	<u>73.5</u>

As evidenced in Table 2, prompts substantially increased the accuracy of few-shot learning methodologies, particularly against well-regarded benchmarks like GLUE and SuperGLUE. Additionally, human-generated prompts significantly improved model performance, making them nearly as accurate as human specialists. Despite this, the study had its limitations, as the small sample size raised concerns about the applicability of their methodology to a broader demographic of language learners.

2.3 Telemetry Analysis Modelling

Based on telemetry data, Remonda et al. (2021) investigated the potential of deep reinforcement learning algorithms to optimize racing car performance. A specialized physics engine designed

for racing simulations was used to capture 7,000 laps on the Aalborg circuit at a rate of 1000 Hz. The dataset included vital channels, capturing details such as throttle and brake inputs, speed, distance covered, and the angle between the car's velocity vector and the track's tangent. Several reinforcement learning techniques were applied, each with distinct hyperparameters and emphasizing experience replay and multi-step targets. Look-ahead of curve (LAC) method was incorporated to enhance model accuracy. This method allows the model to better estimate track curvatures and improve lap times.

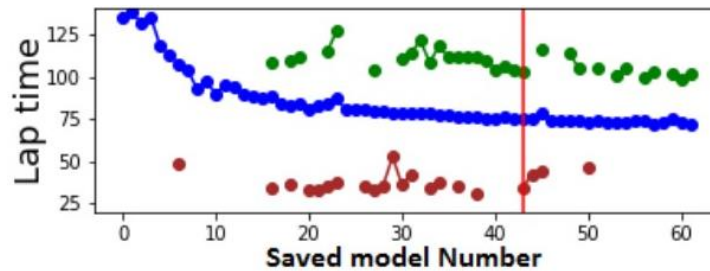


Figure 3: Fastest laps for all ML models on Aalborg (Blue), Michigan (Brown) and Forza (Green) circuit (reproduced by Remonda, et al., 2021).

Figure 3 illustrates the lap times produced by the ML models. Each model was trained exclusively on the Aalborg track and tested on three different circuits. In comparison to standard open-source bots, the LAC-enhanced model reduced lap times by 2.288 seconds in Forza, 0.628 seconds in Michigan and 3.92 seconds in Aalborg. On unfamiliar tracks, however, the ML algorithm experienced difficulties due to overfitting. Recognizing that standard driving patterns could not fully navigate the unique intricacies of each circuit, the integration of a richer telemetry dataset was proposed. Moreover, merging diverse data streams, especially pairing visual imagery with telemetry data, emerged as a promising strategy for bolstering the model's proficiency and adaptability (Remonda, et al., 2021).

Peters and Gerdes (2021) study shifted the focus from analyzing driver behavior to analyzing intricacies of driver behavior using ML techniques. An essential part of their research was ProMoD. A modular framework that mimics human driving actions by using ProMPs for driving line distributions and a recurrent neural network trained from real-world and simulated telemetry data. Displaying its superiority, ProMoD consistently outperformed traditional models like end-to-end supervised learning and DAgger. It showcased remarkable abilities, from navigating unfamiliar tracks to mimicking human-like controls and evolving through experiential learning.

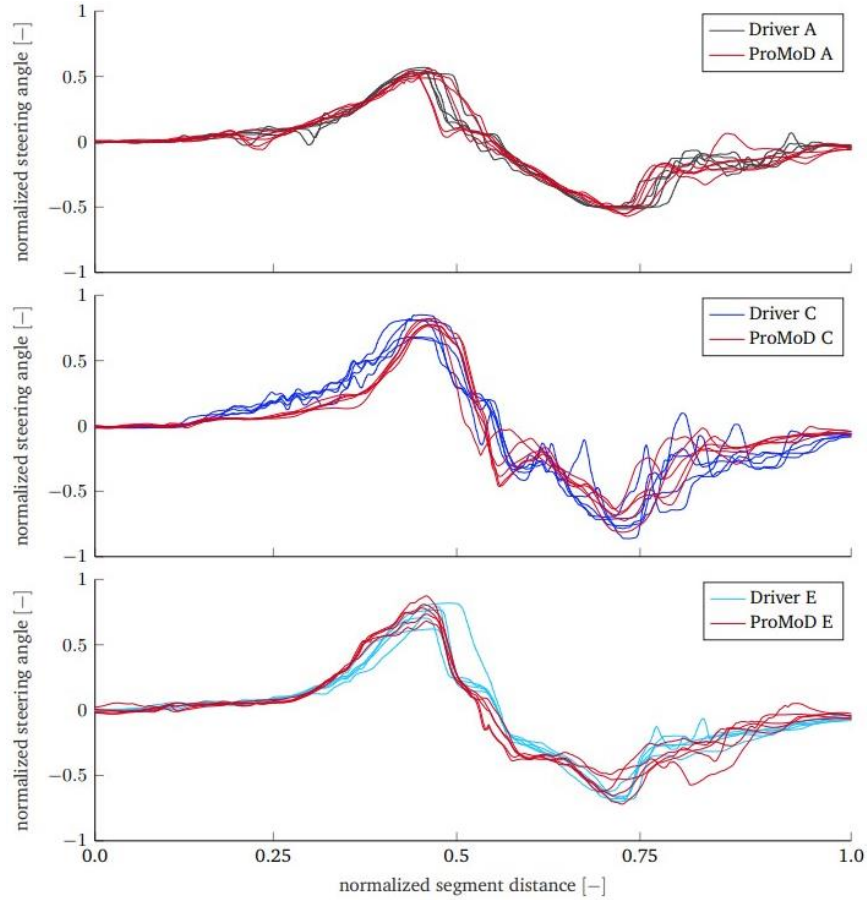


Figure 4: Comparison of steering behavior between ProMoD models and real drivers (reproduced by Peters and Gerdes, 2021).

A refined clustering algorithm, DIMRA, based on decision trees was introduced to decode driver behavior complexity. As shown in Figure 4, DIMRA has the ability to transform vast databases of driver data into comprehensible, actionable patterns. When paired with ProMPs, this synergy supercharged ProMoD's operational efficiency. Consequently, the predictive ML model had an accuracy level of 98.25% across 56 metrics. Despite this groundbreaking success, the study also emphasized the need to further examine ProMoD's adaptability across a range of vehicle configurations (Peters and Gerdes, 2021).

Continuing the theme of driver behavior and data-driven insights, Schleinitz et al. (2022) emphasized the importance of robust evaluation metrics beyond mere accuracy in lap time forecasting. An LSTM neural network architecture was used to predict lap times using data from a driver-in-the-loop simulator. The collected dataset contained 1,226 laps, each with 2,000 data points, allocated to training (70%), validation (15%) and test (15%) sets. In order to obtain optimal

ML performance, the number of LSTM cells, hidden layers, and learning rate were adjusted through a systematic grid search method. A further analysis of the methodology revealed that professional drivers consistently displayed smoother inputs and utilized tire grip potential to a greater degree than amateurs. Incorporating this pivotal insight into the LSTM framework, the study distinguished these nuanced differences, forecasting lap times with twice as high accuracy for both driver categories.

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad \textbf{Equation 2: Mean squared error (Schleinitz, et al., 2022).}$$

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t| \quad \textbf{Equation 3: Mean absolute error (Schleinitz, et al., 2022).}$$

A high accuracy metric serves as a preliminary indication of a model's effectiveness; however, it does not fully capture its robustness. In other words, even a highly accurate model based on familiar data may be susceptible to overfitting, which might impair its performance when applied to new or unfamiliar datasets. Due to the dangers associated with over-reliance on accuracy alone, the researchers considered the metrics of mean squared error (MSE), outlined in Equation 2, and mean absolute error (MAE), specified in Equation 3. The MSE measures the average squared difference between predicted and actual values, providing a measure of the model's overall prediction accuracy. Alternatively, MAE calculates the average absolute difference between predictions and actual values in to provide a more intuitive expression of the model's prediction errors. The LSTM model's performance metrics showed an MSE of 0.04 seconds and a MAE of 0.16 seconds. These results were considered commendable, but their efficacy had been tested primarily within the confines of simulators. Thus, its application in broader, real-world driving scenarios warrants further investigation (Schleinitz, et al., 2022).

2.4 Performance Optimization

Drawing from Schleinitz et al. (2022) study on comprehensive ML predictive model evaluation techniques, Pontin (2023) reviewed classification assessment techniques by developing an AI-based race strategy assistant for Formula 1 at the Paul Ricard circuit. The assistant was composed of two separate neural network architectures to assist in tire and pit stop decision making. The pit stop decision neural network was structured with a single hidden layer and a

sigmoid output layer. It featured an input layer of 10 nodes, representing the car and track data, and an output layer dedicated solely to the pit stop decision. The pit stop predictive model was trained on an imbalanced dataset of over 3.6 million labeled data points. To counter this bias, a class weight was added to enhance the significance of the "pit stop" category, and the model boosted its accuracy to 99.7%. Alternatively, the tire decision neural network was trained on an expansive 1.5 million data point dataset using a 12-node input layer corresponding to track and tire data and four nodes output layer corresponding to a tire choice.

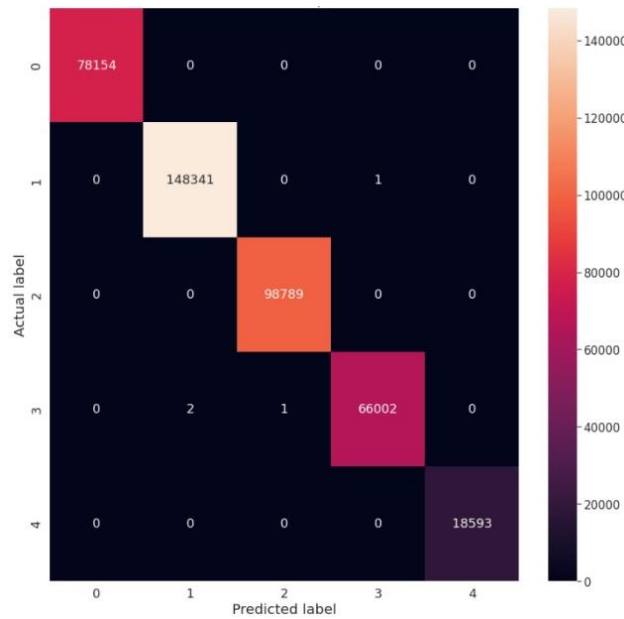


Figure 5: Number of samples of the class predictions made by the tyre deciding NN on the test data (reproduced by Pontin, 2023).

For addressing the complexity of selecting the correct tire compounds during pit stops, confusion matrix methods were employed, as shown in Figure 5. For each category within the dataset, this method identifies true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Owing to its precise classification capabilities, the tire decision model scored noteworthy 99.9% accuracy. To further refine the system's prediction accuracy and realism, the author suggested the integration of richer real-world data, supplemented with increased data points and refined sensor accuracy (Pontin, 2023).

In line with the focus of this master's thesis, Hojaji et al. (2023) applied non-linear ML algorithms to predict lap times using race car telemetry. The dataset, comprising 300 laps from 10 drivers (30 laps each), incorporated key channels such as speed, throttle, brake, gear, engine RPMs,

and steering angle. The data was split, with 70% used for training and 30% for testing across three distinct ML algorithms.

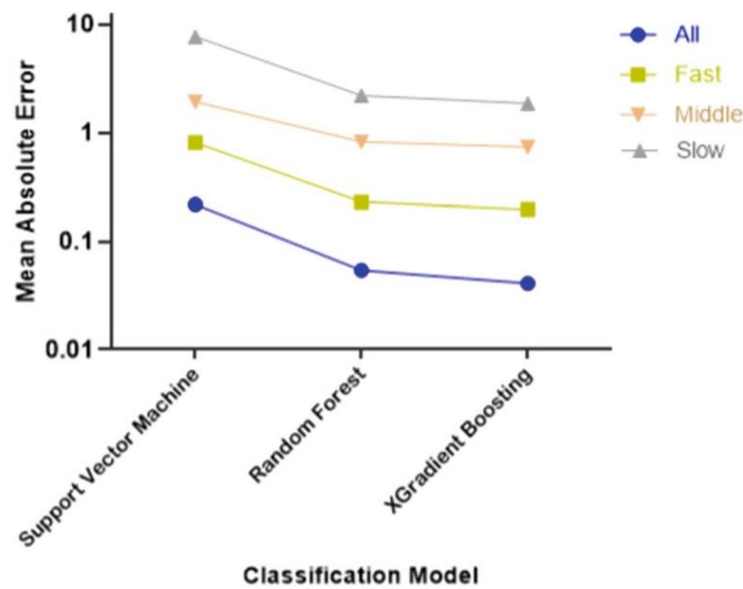


Figure 7: MAE of the tested ML lap time predictive models (reproduced by Hojaji, et al., 2023).

Figure 6 shows that the XGradient Boosting model stood out, predicting lap times with an absolute accuracy of 92.2% and an error rate of 7.8%. As part of the model optimization process, hyperparameter tuning was conducted using grid search, emphasizing parameters like learning rate, maximum depth and number of estimators.

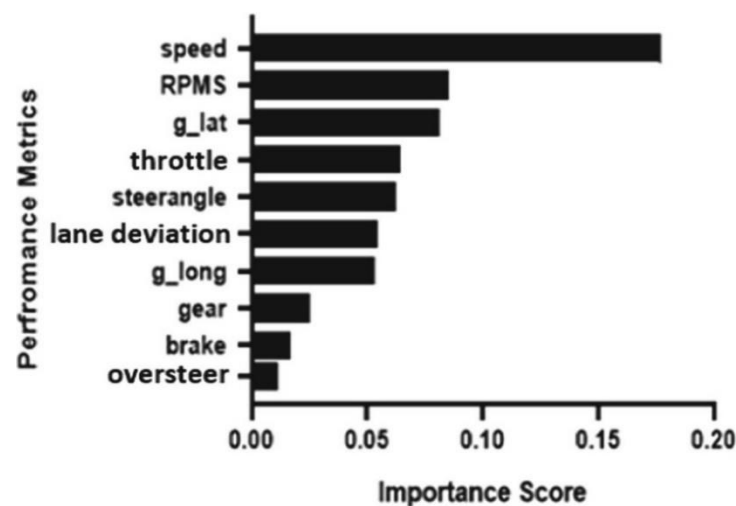


Figure 9: Importance of analyzed channels on lap time (reproduced by Hojaji, et al., 2023).

To further sharpen the model's predictive power, a dual-step refinement was added. Initially, statistical measures, including mean, median, max, and standard deviation, were deduced for each channel, offering insights into the data distribution. This was followed by a backward elimination technique to determine the relevance of individual features. Such a nuanced strategy allowed the algorithm to pinpoint correlations between telemetry channel significance and lap time, Figure 7. As a result, the model displayed an MAE of 0.15 seconds, an MSE of 0.04 seconds and an R^2 score of 0.98. The R^2 score is commonly used in non-linear relationships, and it shows how closely the actual values correspond to the model's predictions as 1 indicates a perfect match. Nevertheless, the authors suggested that a segmented analysis of the circuit might offer even finer precision, enabling the model to focus on shorter timeframes rather than an entire lap (Hojaji, et al., 2023).

3 Methodology

Throughout this project, Python 3.11 was selected as the cornerstone of a programming framework for implementing and documenting these methodologies. A total of 14 notebooks were meticulously constructed, serving as encapsulated environments for integrating code, visualizations and explanations. Each notebook was tailored to address specific problem-solving strategies, fostering a structured and organized approach. This section critically analyzes these approaches to shed light on their particular nuances and contributions.

3.1 Data Collection

Multiple studies, as outlined in Chapter 2, underscore the significance of iterative training for ML models. As a first requirement, consistent datasets had to be deployed to establish a baseline for the telemetry model. Harmonizing uniform circuit configurations and motorcycles enhances the model's training and allows more precise emulation of target performance achievements.

Table 5: Overview of all collected telemetry racing data.

Circuit	Bike	Laps
Brands Hatch	R6	66
Donington GP	R6	72
Donington National	R6	94
Mugello	R6	179
Oulton Park	R6	89
Pembrey	R6	142
Phillip Island	R6	53
Snetterton 200	R6	128
Spa Francorchamps	R6	110
Aragon	ZH10R	150
Phillip Island	ZH10R	834
Donington GP	Panigale	96
LeMans	Moto2	183
Qatar	Moto2	178

The data used for this project originates from the Phillip Island circuit and revolves around a Kawasaki ZX10R chosen for its alignment with the project's requirements, Table 3. The dataset was provided by Andrew Harrison, a data engineer associated with “Intermoto Ponyexpres”, a noteworthy competitor in the FIM Superbike World Championship (WSBK) SuperSport category (FIM, 2015). Spanning from 2012 to 2015, this comprehensive data encompasses complete race weekends and dedicated test sessions. During this period, four distinct riders participated under

the team's banner. An important aspect of this dataset was the inclusion of setting sheets within the 2015 data subset. As outlined in Section 1.2.2, this lays the foundation for evaluating the ML predictive model's effectiveness.

3.2 Data Preprocessing

There are two distinct categories of telemetry data channels. The first encompasses main channels, which are directly measured through dedicated motorcycle sensors. The second category includes calculated channels, which are derived from one or more main channels through mathematical calculations or algorithms. As emphasized by Trapani (2021), the pursuit of optimal lap times begins with a comprehensive exploration of the following channels:

- Engine's rotational speed, measured in revolutions per minute (rpm), provides valuable insights into acceleration, gear-shifting strategies and overall engine performance.
- Transmission's exact gear position influences acceleration, responsiveness and overall performance by guiding optimal engagement points.
- Wheel slip, calculated as the percentage difference between front and rear wheel speeds, serves as a guide for balancing grip and bike control during throttle and braking modulation.
- Correlated speed is displayed via GPS data in kilometers per hour (km/h), and serves as a linchpin for evaluating acceleration, braking efficiency and overall lap times.
- Lean angle, measured in degrees from a perfectly horizontal orientation, plays a significant role in determining cornering speeds, stability and overall control during turns.
- Front and rear suspension positions in millimeters (mm) shed light on the bike's response to track irregularities, allowing for potential fine adjustments to harness an equilibrium between comfort and performance.
- Throttle position, expressed as a percentage, allows the evaluation of acceleration profiles and power distribution.
- Front brake pressure, usually expressed in bar, allows for optimizing braking points and stability during deceleration.

The selected telemetry data was reviewed using MoTeC i2 Pro software to ensure that all important channels had been captured. While the lean angle channel was not always recorded, it was decided not to include it in the ML dataset. The remaining channels were meticulously charted within a single workspace. Then, each session was exported as separate comma-separated

values (CSV) files. Leveraging CSV format proved beneficial due to its widespread compatibility, simplicity and adaptability across various ML environments (Aggarwal, 2018).



Figure 14: Potential issues associated with in-and-out laps.

When exporting data from the MoTeC telemetry software, an option exists to exclude sessions' in-and-out laps. Figure 14 illustrates the difference between an in-lap, shown in color, and a standard lap, shown in white. The reason for this distinction is rooted in safety protocols that account for pit garage entries and exits (FIM, 2015). Consequently, including such laps in the dataset could potentially induce the ML algorithm to discern patterns prone to overfitting. As a compromise, these two laps were excluded from each session when exporting the data.

The exported dataset followed a tabular structure, with data points being updated at 0.05-second intervals. There was a dedicated column for each of the eight channels, as well as one for time and distance. The time and distance values, however, progressed continuously throughout the session, making it impossible to distinguish between laps. An effective solution was developed utilizing Python 3.11. It focused on extracting and converting the time column into floating-point numbers. Computer programming relies heavily on floating-point numbers, particularly for calculations requiring precise values across a wide range of data. Their finite precision can sometimes lead to rounding errors. Nevertheless, compared to alternative methods like fixed-point or arbitrary-precision arithmetic, floating-point numbers proved optimal, particularly when

managing frequent measurement updates (Versloot, 2019). The transformed dataset was then segmented into laps using cumulative lap times provided by the MoTeC package. Each lap was represented by a new DataFrame, which were seamlessly combined into a unified dataset. In addition, the function's validation protocol included the identification of not a number (NaN) values across all channels. This comprehensive approach effectively omitted laps influenced by sensor irregularities, contributing to the overall dataset's credibility. Underpinning these accomplishments are the Python libraries Pandas and Scikit-learn (NumFOCUS, 2023), (Scikit-learn, 2023).

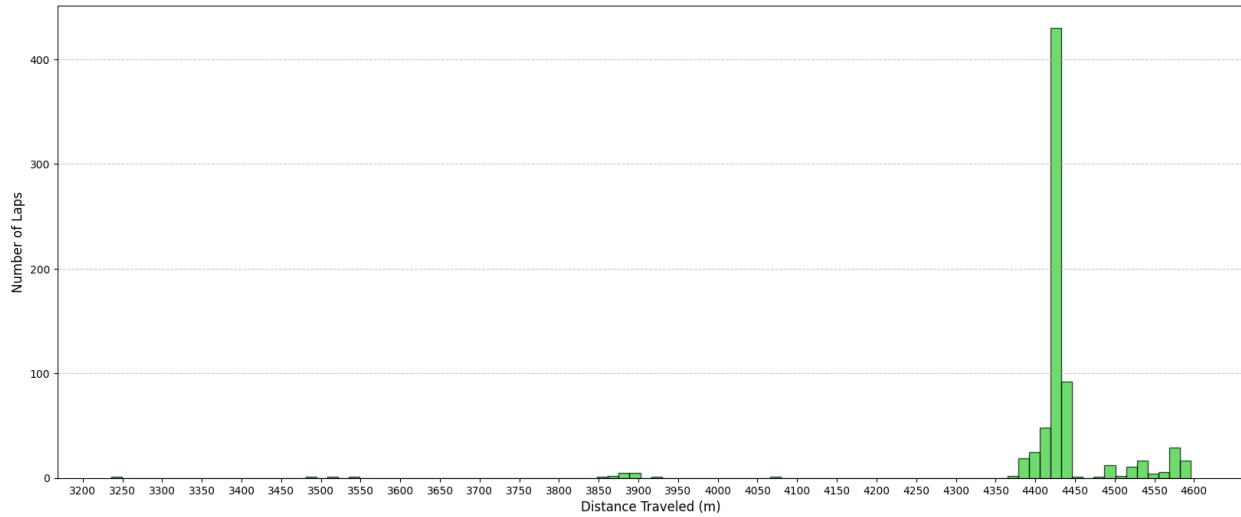


Figure 15: Distribution of GPS distance traveled across all analyzed laps.

Phillip Island circuit covers a total distance of 4,430 meters (Phillip Island Circuit, 2023). However, there have been instances when lap times appeared to defy realism. Upon careful review of the MoTeC data, this anomaly was traced to the GPS failing to capture the final 1,000 meters of the track, as shown in Figure 15. An additional preprocessing function was integrated. Considering an estimated 10% variance in covered distance due to riders adopting diverse racing trajectories. This function effectively excluded laps outside the range of 4,400 and 4,600 meters. The extension of the upper limit was based on the understanding that a longer traveled distance may result in a slower lap time due to an error by the rider. This type of occurrence may contribute to the ML model's learning curve, enabling it to recognize and mitigate errors in its prediction capabilities according to Pontin (2023). Overall, the data cleaning process resulted in the omission of 230 laps. Leaving 490 laps for training and 120 for testing based on the 80-20% allocation principle referenced in Section 2.1.

Telemetry channels can have different scales or units of measurement. Normalization addresses this by converting the values into a standard scale without distorting the difference between the ranges. For ML, such standardization is imperative. It ensures that individual channels do not exert a significant influence over results due to their sheer size. Consequently, normalization enhances feature scaling, accelerates convergence and reduces outliers (Scikit-learn, 2023)

$$\text{Normalized}(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad \text{Equation 4: Normalization (Scikit-learn, 2023).}$$

The z-score and min-max scaling are two popular methods of normalization. In the first case, data is centered around zero without ensuring a range is bound. In contrast, the latter approach, as illustrated in Equation 4, usually scales channel values between 0 and 1. For this research, min-max scaling was adopted. By ensuring that all telemetry data channels are confined to a uniform scale, direct comparison and analysis becomes easier according to Remonda (2021).

The final preprocessing step consisted of partitioning the data into predictive features (X) and corresponding targets (Y) using the function "separate_features_target ()". The distinction created clear boundaries for training and facilitated optimized evaluation. Furthermore, by isolating these features, adjustments, scaling or transformations may be applied without compromising the integrity of the target data. Thus, the model's capability to predict the target is enhanced, yielding trustworthy and interpretable results during both the training and evaluation phases (Kuhn & Johnson, 2019).

3.3 Feature Engineering

Once a reliable dataset was established, the focus shifted to feature engineering. This phase involves modifying or creating new variables derived from the initial dataset. The primary objective is to extract additional insights to broaden the predictive features (X). Thus, the ML algorithm will have a more comprehensive set of patterns to learn from (VanderPlas, 2016).

In Section 2.3 of the literature review, Recursive Feature Elimination (RFE) was highlighted as a prevalent feature engineering method, especially when training ML models on telemetry data. RFE is a systematic approach for identifying the most significant features of an ML model. It initially uses statistical metrics, such as the mean, median, maximum, and standard deviation, to

determine how data is distributed across different channels. In subsequent iterations, a model is trained, features are ranked based on their significance and the least crucial ones are systematically removed. This process continues until a specified number of features remain or certain criteria are fulfilled.

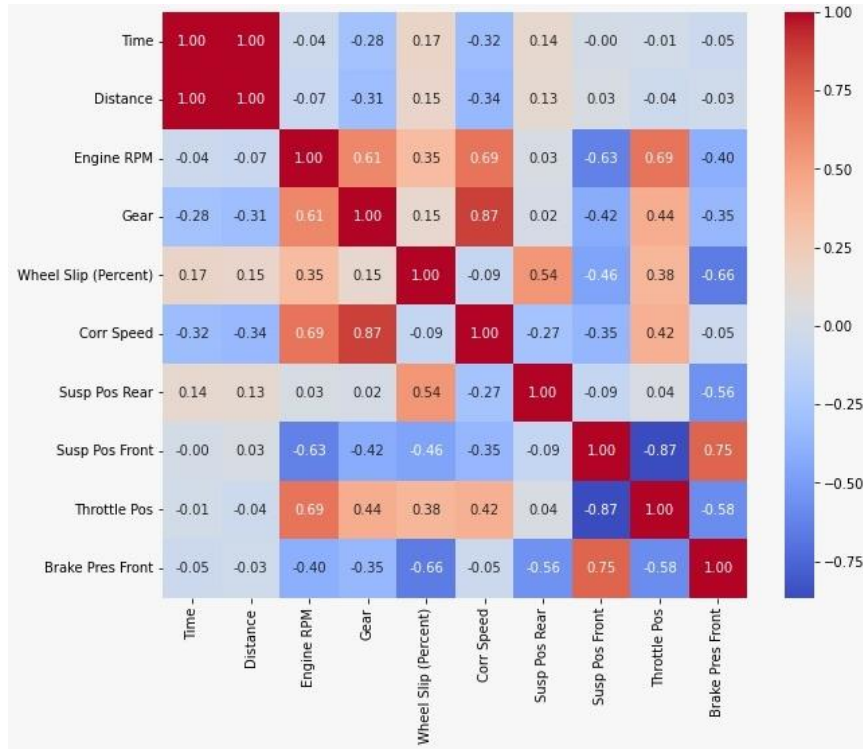


Figure 16: Average correlation matrix for all channels across all laps.

A positive correlation coefficient indicates that both variables move in tandem, increasing together. Conversely, a negative coefficient suggests the variables move in opposite directions. For instance, in Figure 16, the heatmap reveals a robust positive correlation between correlated speed and gear position channels. This was deemed logical, as a higher gear corresponds to a greater gearing ratio, leading to increased top speed (Bradley, 2018). A correlation such as this may assist the model in grasping motorcycle dynamics with greater accuracy, aligning closer to human perceptions of how motorcycles operate, thereby improving its predictive ability.

Throughout the project, the RFE method consistently demonstrated efficiency. While several methods were attempted during the development phase, with varying degrees of success, they are discussed later in this chapter to maintain narrative continuity.

3.4 Development

Based on the literature review, decision tree models were considered necessary to effectively capture the intricate patterns within the telemetry data due to its inherent non-linearity. This necessity aligned with the essence of both XGBoost and Random Forest, prominent ensemble learning algorithms. RF combines multiple decision trees independently and aggregates their predictions, introducing randomness as a result of feature selection and data subsets. In contrast, XGBoost uses a sequential boosting mechanism, which refines predictions by iteratively training new trees. Generally, RF can better handle non-linearity and noise by taking advantage of averaging, whereas XGBoost can capture intricate patterns by boosting (Gupta, 2021). As compared by Hojaji et al. (2023) in Section 2.4, the XGBoost model exhibited a slight advantage, surpassing the RF model only by 2% when forecasting lap times through race car telemetry. This project, however, aimed to reverse the process and predict channels' performance based on lap times. As the given dataset contained many unknowns such as weather conditions, riders and chassis setups, the RF model was chosen due to its robustness to noise and its ability to handle diverse and inconsistent data (Scikit-learn, 2023).

3.4.1 Lap Time Prediction Model

In a many-to-one predictive model, multiple attributes are combined to forecast a single outcome. This methodology is more streamlined in identifying underlying patterns than predicting numerous outcomes from a singular value (Aggarwal, 2018). In light of this, it was decided to begin the modeling phase by developing a many-to-one predictive algorithm capable of forecasting a lap time (outcome) using multiple performance channels from the collected data. In the first step, the values from these eight selected channels were processed and averaged to form predictive features (X), Section 3.2. The consolidated feature set was then used to calculate the target lap time. Mirroring Hojaji et al. (2023), the model used "GridSearchCV" to fine-tune its hyperparameters. In essence, the configuration is automatically adjusted to achieve the optimal balance between accuracy and convergence. These essential parameters included:

- Number of trees (n_estimators).
- Depth of each tree (max_depth).
- Minimum data points a node requires for further branching (min_samples_split).
- Least data points for a node to finalize a decision without branching (min_samples_leaf).

For the purpose of evaluating the model's performance consistency on unseen data, the model was cross-validated (cv) ten times. This widely-accepted approach segmented the dataset into ten portions. During each iteration, the model was trained on nine of these segments while the tenth served as the validation set (Scikit-learn, 2023). The objective of this meticulous validation process was to achieve an R^2 score of 0.95. If the average score of the lap time prediction model did not meet this threshold, the hyperparameter scope was adjusted using the “update_param_grid ()” function. A technique paralleled with the learning rate range test detailed in Section 2.1.

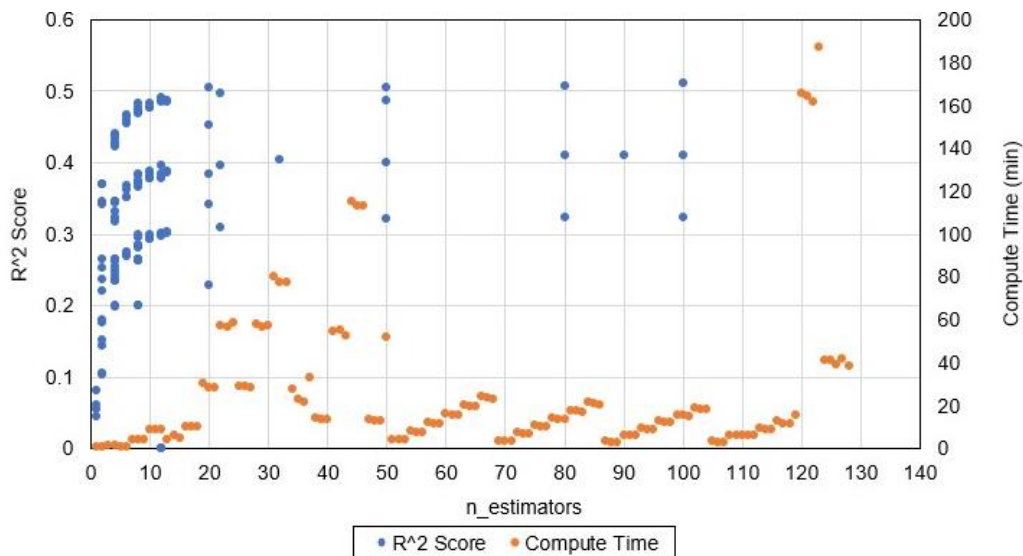


Figure17: Initial hyperparameter tuning process of the Random Forest-based lap time prediction model.

Approximately 1,000 configuration settings have been identified for the initial ML model. There was a systematic cross-validation test conducted ten times per setting, resulting in the assessment of around 10,000 iterations. Figure 17 illustrates the model's performance in relation to the number of trees. Despite these rigorous efforts, the optimal R^2 score achieved was 0.510. The result indicated that the model either failed to grasp the dataset's intricacies (underfitting) or was too specialized for the training data (overfitting). Upon evaluation, it became evident that the model treated each lap as an isolated event. In practical scenarios, however, the outcome of one lap can significantly influence the outcome of the next. A poor exit from the penultimate corner can result in a reduced top speed on the start-finish straight, affecting the performance of the next lap. Furthermore, different sessions may exhibit distinct characteristics due to differences in riders, weather conditions or mechanical setups.

As a result, the algorithm was configured to detect instances where the “Distance” value resets or drops from its previous value. In such situations, a new session is deemed to have begun. As a result, each session was assigned a unique "SessionID". A sequential number was also assigned to each lap within these sessions. These additional steps were ultimately incorporated into the model’s features (X). Using the same GridSearchCV procedure as before, the ML model demonstrated notable improvements. It not only achieved R^2 score of 0.948 but reduced its processing time by around 200% and its memory usage by 85%. Following the training phase, it was imperative to make the model readily available for future applications. To ensure consistency of the model's architecture and state, it was serialized in a pickle format (Python, 2023). Lastly, the model was assessed on randomly chosen unseen data to confirm its integrity and validity.

Table 7: Random Forest-based lap time ML predictor performance after advanced preprocessing.

Session	Actual Lap Time	Predicted Lap Time	MAE	PE
	sec	sec	sec	%
GP_FP1_EX1_4Laps	95.876	95.499	0.377	0.393
GP_FP1_EX2_4Laps	95.876	95.505	0.371	0.386
GP_FP2_EX2_4Laps	96.085	96.011	0.074	0.077
GP_FP2_EX3_4Laps	96.085	96.073	0.012	0.013
Test1_Afternoon4_9Laps	96.472	96.342	0.130	0.134
Test1_Afternoon5_12Laps	96.472	96.385	0.087	0.091
Test2_Afternoon3_14Laps	94.025	95.406	1.381	1.469
Test2_Afternoon4_13Laps	94.025	95.451	0.426	1.516
Test2_EX2_5Laps	96.417	96.315	0.102	0.106
Test2_EX3_8Laps	96.417	96.302	0.115	0.119
Average:			0.307	0.430

The lap time model achieved an average MAE of 0.307 seconds, accompanied by a percentage error (PE) of 0.430 %, as shown in Table 4. PE is another relevant ML metric, which quantifies the disparity between predicted and observed values. A PE of 0 indicates perfect accuracy, indicating no variance between predictions and actual results (Aggarwal, 2018). It was observed that there was room for even finer results through subsequent hyperparameter optimization. Even so, the ML model was considered a resounding success due to its ability to forecast lap times within a narrow range of 0.5 seconds.

3.4.2 Telemetry Prediction Model

In its initial phase, the algorithm was capable of accurately predicting lap times (outputs) based on telemetry data (inputs). The ultimate objective however, was to compute precise sensor

readings from a specific lap time (output). Thus, allowing data engineer to highlight performance gaps in comparison with the original rider's telemetry data. To accomplish this, it was necessary to reverse-engineer the lap time prediction ML model to generate inputs from outputs. The conventional approach would have been the employment of autoencoders. They are intricate deep learning constructs that are designed to identify complex relationships among data. These relationships are then compressed into a condensed latent space before the original dataset can be reconstructed. Due to the complexity of tuning hyperparameters and the computational demands of deep learning models, it was decided to continue using the random forest regressor (Aggarwal, 2018). This revised model deviates from convention by repurposing the "Time" column. Instead of being treated as a dependent variable (output), it served as a basis for the development of other variables. Therefore, by setting parameters such as time and distance, the model can project behavior across various telemetry channels.

A set of conditions was formulated. Firstly, a benchmark lap time, sourced from the record of 93.072 seconds, established in 2018 by Sandro Cortese on a Yamaha R6 in the WSBK SS class (Phillip Island Circuit, 2020). The distance according to this timeframe was set to the total track length of 4,430 meters. Lastly, data point predictions were regulated by a time interval of 0.5 seconds. This interval was determined to offer the most lucid interpretation of the channels. A high interval may obscure critical details by oversimplifying the trace, whereas a low interval could introduce undue noise and obscure vital elements. Refer to Appendix D for an in-depth examination of this decision.

Once the outputs were defined, the slightly modified ML model was trained with the exact same estimator hyperparameters as established in the lap time prediction model. Unfortunately, this strategy was inefficient and memory-intensive. In recognition of the previous success of feature engineering in enhancing the efficiency of this phase, a polynomial regression model has been seamlessly incorporated into RF framework. This mathematical model allows variables to be exponentiated and then multiplied by coefficients. In conjunction with the "polyfit" function, the model can determine the optimal coefficients for a polynomial that best represent the data. Due to the unique fits provided for each channel as a function of time, this hybrid ML model should have become more computationally efficient and accurate (NumPy, 2023).

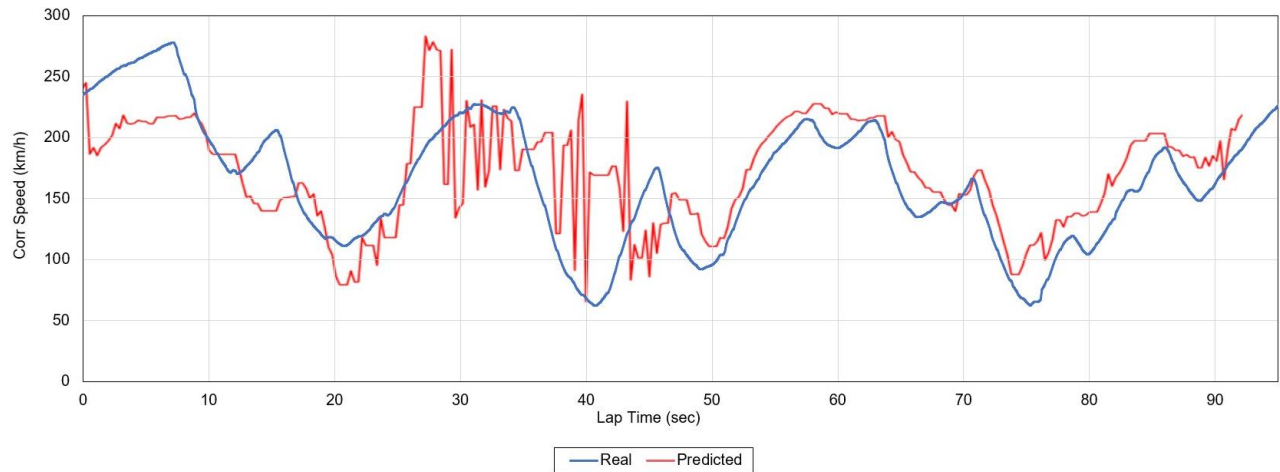


Figure 18: Correlated speed comparison between real and predicted using a hybrid ML model.

Following rigorous hyperparameter optimization, the hybrid model achieved its best score on the correlated speed channel with an MSE of 2,013.83 and an MAE of 87.01. These metrics were deemed excessively high. Figure 18 illustrates the model's attempt to forecast the rider's exact speed needed to surpass the circuit record with a random sample of unknown data. It was evident that the model's predictions and the actual telemetry data were correlated. Nonetheless, the model's reliability was compromised by the pronounced fluctuations in acceleration and deceleration. A key factor contributing to this issue was the incorporation of polynomial regression. Approximately 45 minutes were required for the model to converge for each configuration, suggesting an overcomplexity caused by high coefficients in the polynomial equations.

Consequently, the polynomial regression element was removed. The focus shifted towards segmenting the track into distinct sections. Phillip Island circuit consists of four sectors (Phillip Island Circuit, 2023). By dividing the track, granular modeling was intended, encapsulating the unique attributes and challenges of each sector. In addition, modeling smaller segments could reduce the complexity of the task, thus potentially minimizing overfitting that might occur when modeling the entire circuit.

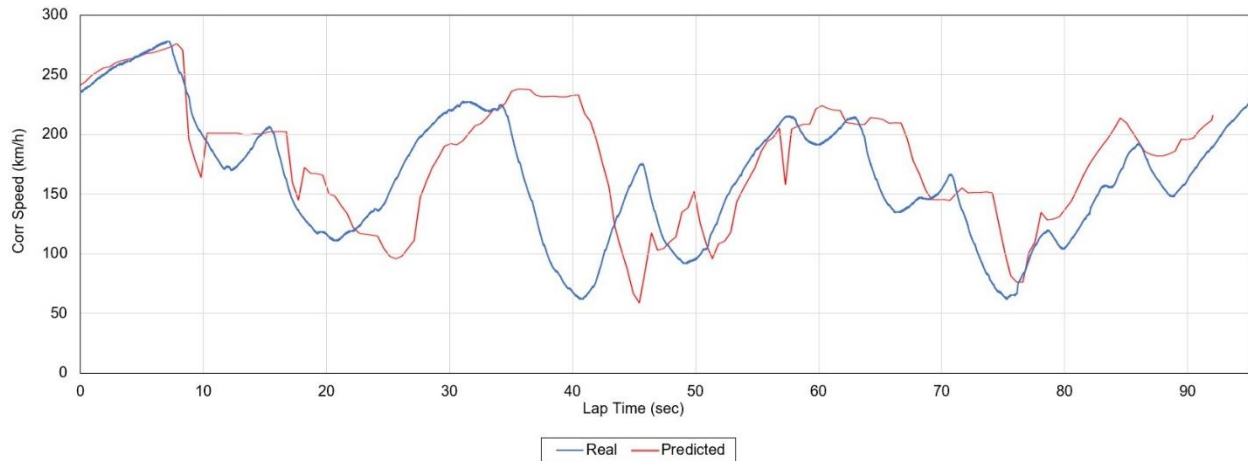


Figure 19: Correlated speed comparison between real and predicted using feature engineering to split the circuit into sectors.

The revised model demonstrated a significant improvement in computational efficiency, as shown in Appendix C. To further speed up the optimization stage, data sampling was incorporated. Based on this approach, only 20% of the dataset was randomly selected for training the model. Accordingly, the ML model generated an MSE of 0.00 and MAE of 0.03 for the corrected speed channel. Despite these near-perfect scores, anomalies surfaced in the telemetry predictions, as illustrated in Figure 19. The model frequently recommended braking much later than was practical. In theory, this strategy would enable the rider to maintain higher speeds through the corners and thereby improve lap times.

It was, however, the engine speed channel that presented the most persistent performance problems. This channel recorded a notable MSE of 538,569.70 and a MAE of 469.55. The behavior of the predicted engine speed appeared disjointed, lacking both coherence and recognizable trends. To address this issue, the circuit was further segmented into straight and corners, as detailed in Appendix C. This was achieved by using one-hot encoding. An advanced feature engineering method that enhances prediction accuracy by converting categorical data into a machine-learnable format. Instead of representing a categorical variable within a single column filled with varying labels, one-hot encoding divides this column into several columns, each with a distinct label (Scikit-learn, 2023). Using this approach, the prediction accuracy was significantly improved across all channels. In particular, the engine speed prediction performance improved by approximately 98%. Despite this, the results were still not optimal.

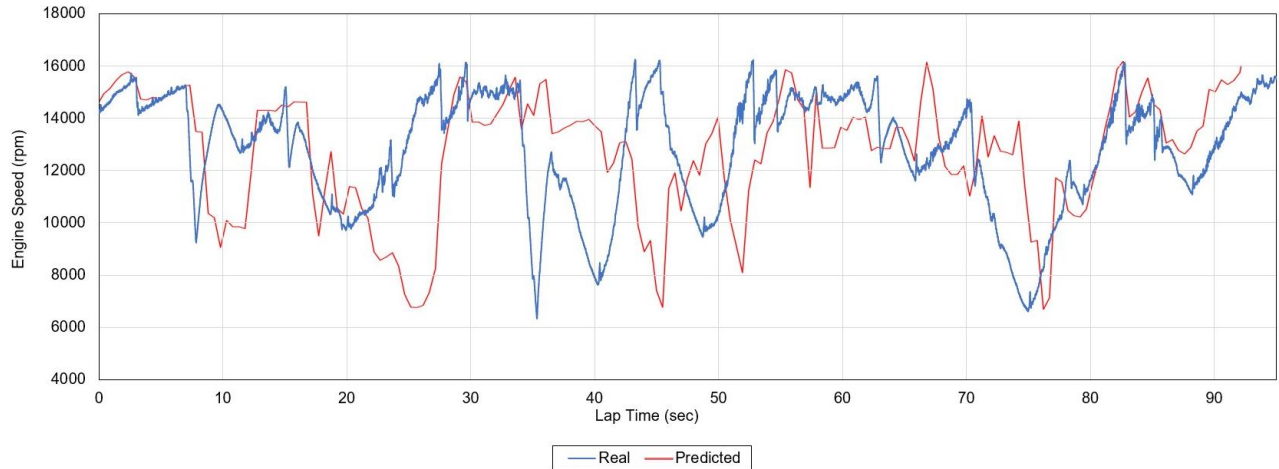


Figure 20: Engine speed comparison between real and predicted using advanced feature engineering to split the circuit into straights and corners.

The predicted behavior of the engine speed channel substantially deviated from the actual observations, as shown in Figure 20. An in-depth analysis resulted that the primary impediment to achieving accurate engine speed predictions was the intrinsic noise within the channel. This highlighted the need to identify an additional relationship within the model that can guide its decisions. Consequently, a RFE experiment was undertaken to potentially isolate these areas for improvement.

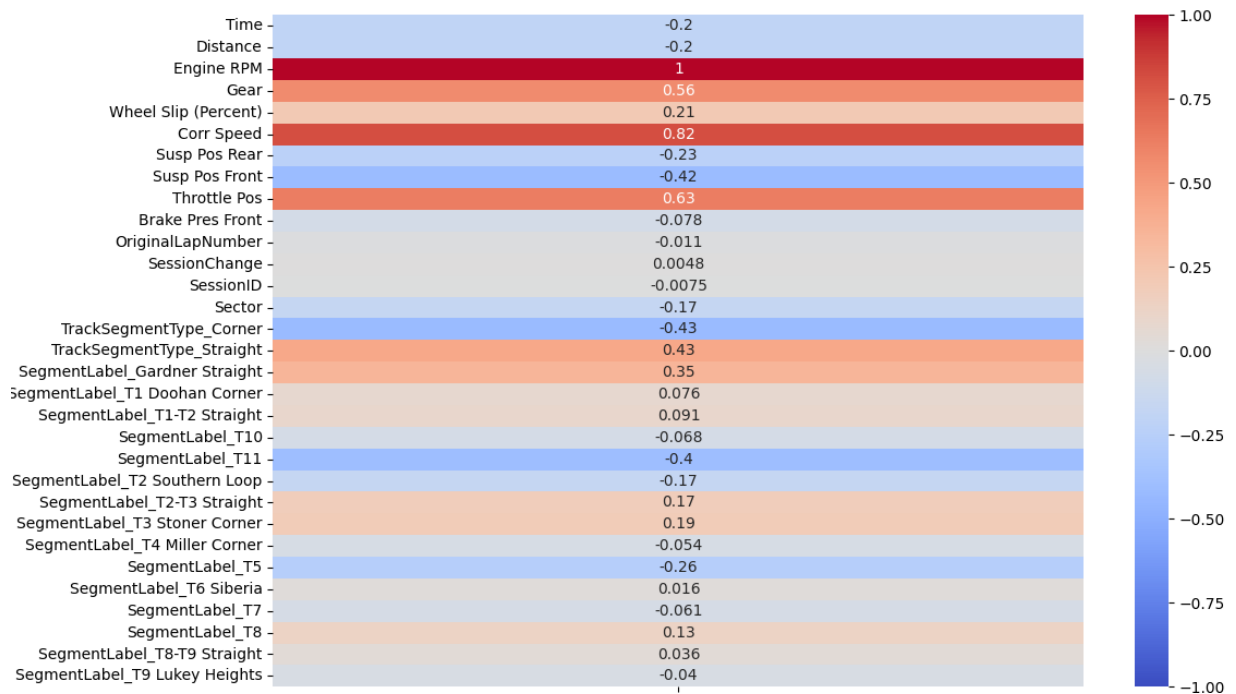


Figure 21: Correlation matrix of engine speed across all laps.

Figure 21 shows that correlated speed, gear and throttle position were the variables with the most substantial correlation coefficients in relation to engine speed. Traditionally, increasing throttle input results in a surge in power output, which in turn revs up the engine. In response, the speed increases, often requiring a gear shift, according to Bradley (2018). Recognizing the robust relationship between these variables, it was harnessed to enhance the model's efficiency. The telemetry ML prediction model was subdivided. The primary model maintained the latest techniques suitable for all channels. While the secondary model, activated following the primary's convergence, re-trained the correlation speed predictions, incorporating the three most correlated channels into its features (X). The methodological adjustment resulted in a significant 98% performance enhancement.

In the final stage, both models underwent rigorous hyperparameter optimization and training using the available training data at hand. Interestingly, both models converged on identical estimator configurations: “n_estimators” was set to 15, “max_depth” was left undefined (None), “min_samples_leaf” was designated as 1 and “min_samples_split” was established at 2. Appendix E provides a more detailed representation of the trees structure generated by the RF model.

Table 9: Average prediction performance metrics on unseen data.

Channel	MSE	MSE	MAE	MAE
	20% Data	100% Data	20% Data	100% Data
Engine Speed	229062.46	460.20	279.69	10.20
Gear	0.07	0.00	0.07	0.00
Corr Speed	20.28	0.01	3.40	0.04
Wheel Slip	5.62	0.01	1.61	0.04
Susp Pos Rear	18.39	0.01	3.29	0.05
Susp Pos Front	116.88	0.08	6.57	0.05
Throttle Pos	154.16	0.08	6.45	0.05
Brake Front Pos	0.45	0.00	0.29	0.00

Table 5 presents a comparative analysis of the ML model's performance on randomly selected test data when trained on 98 laps versus 490 laps. As anticipated, utilizing the complete dataset significantly enhanced the model's overall performance. This improvement was however offset by a 25% increase in computational resources.

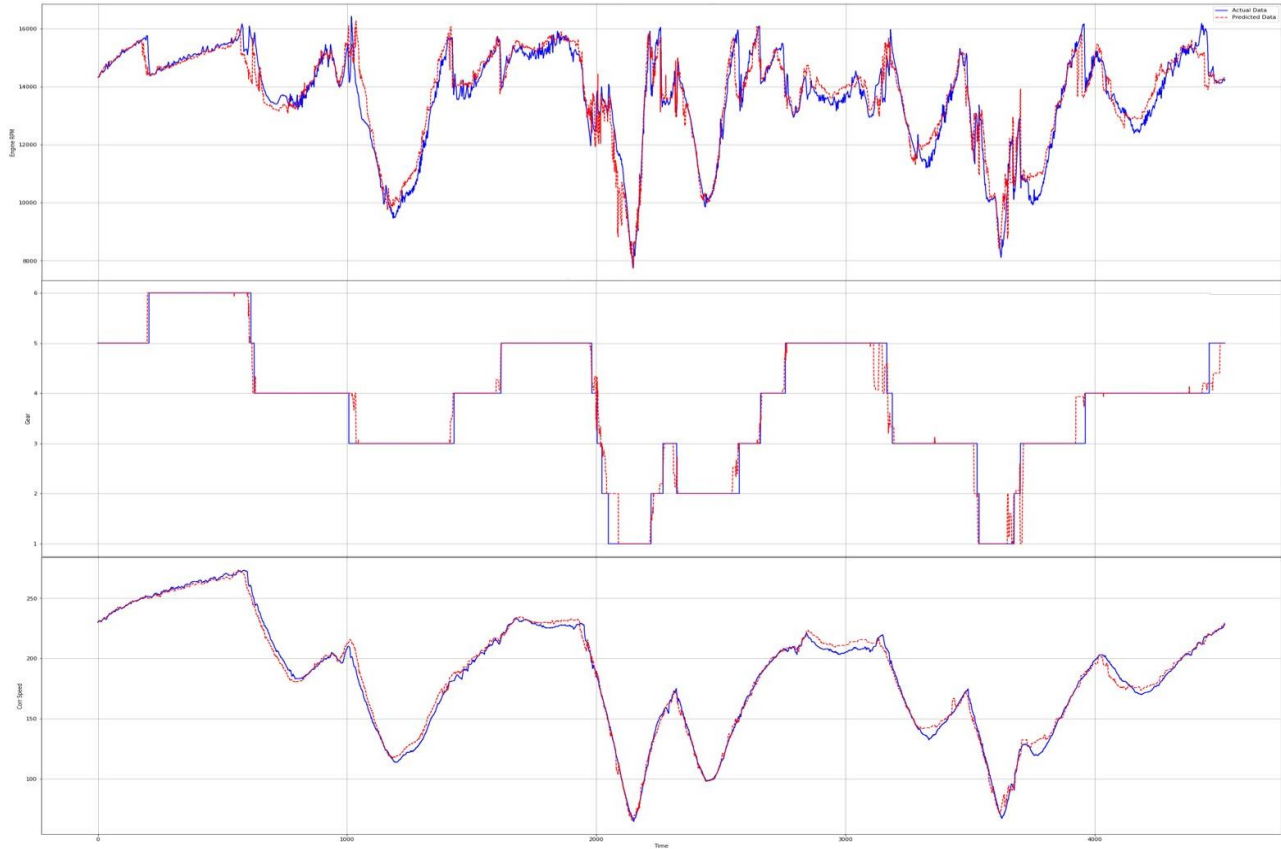


Figure 22: Engine speed, gear and correlation speed channel comparison against randomly selected unseen data without a filter.

Reviewing the results shown in Figure 22, it became apparent that the model adopted tactics similar to those employed by experienced riders (Ibbot, 2021). Maintaining higher speeds through corners by late and sharp braking, which then provided an advantage on straights. Additionally, the model seemed to maintain a slightly higher engine speed in slower corners by either increasing the slip percentage or selecting an alternative gear. By doing so, the engine speed will remain within the optimal torque zone, resulting in faster acceleration out of corners.

According to Figure 22 and Appendix E, the model's performance was commendable. There was however a need for further improvement with regard to the pervasive noise observed across most channels. As an example, the model displayed uncertainty regarding the selection of the optimal gear in specific sectors. To mitigate this, a method known as the "weight of probability" was introduced, as described in Section 2.1. Through the use of this method, the model could gauge its own confidence levels in its predictions, which effectively reduced the amount of noise. In response, the model may display a more refined and human-like behavior on the track.

3.4.3 Chatbot

Chatbots are software applications designed to simulate human conversation through text or voice commands. For this particular project, the chatbot was engineered to assist users in identifying areas for performance improvement through textual guidance. Drawing power from state-of-the-art AI models, chatbots can comprehend complex requests, tailor responses and improve their interactions over time. Several OpenAI models are available to the public and each can be smoothly integrated with the telemetry prediction ML model to produce solutions based on input (OpenAI, 2023). In this instance, users are restricted to only providing telemetry data to the chatbot without engaging with it further. It becomes imperative for the ML model to adeptly translate its findings into a comprehensible written prompt, enabling the chatbot to process it as standard input.

Motorcycle riders exert a major influence on their vehicles, in contrast to car drivers. Even slight changes in a rider's body weight can have a significant effect on the motorcycle's center of gravity (CoG) affecting its performance (Bradley, 2018). Thus, the primary focus should be on improving riding technique. Only after that, mechanical or electronic adjustments can be considered. A practical approach to pinpointing areas for improvement in riding technique is to use an ideal lap as a reference (Segers, 2014). This theoretical lap is derived from the rider's best sectors. As soon as this benchmark is exceeded, attention can be focused on setting a new lap record.

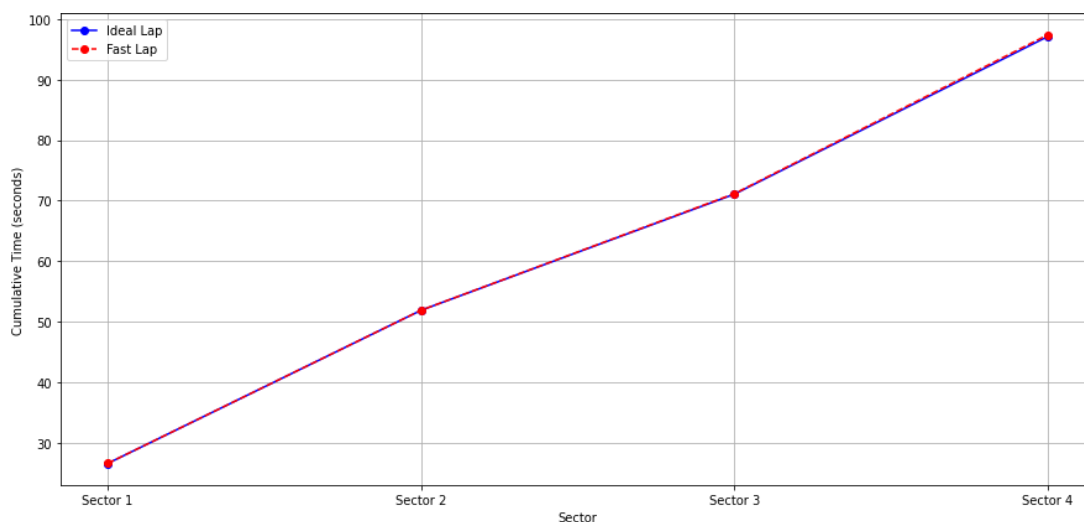


Figure 20: Cumulative time per each sector between ideal and fast lap.

To encapsulate this philosophy, the ML model was enhanced to generate ideal laps based on the sessions it analyzed. Drawing from the MoTeC software, a cumulative time for each track sector

was calculated as shown in Figure 23. This computation made use of the preprocessing floating-point numbers technique. A steep gradient in the graph indicates an error on the part of the rider, whereas a steady increase indicates a broader challenge. Cross-referencing the deduced results with the split reports in real-time revealed a discrepancy of only 0.05 seconds. Appendix E contains the full analysis.

```
[{'Segment': 'T12',
  'Fast Lap Gears used': [3, 4, 5],
  'Ideal Lap Gears used': [3, 4, 5],
  'Fast Lap Max RPM': '15990 rpm',
  'Ideal Lap Max RPM': '15906 rpm',
  'Fast Lap Min RPM': '12288 rpm',
  'Ideal Lap Min RPM': '11922 rpm',
  'Fast Lap Gear Change RPMs': 'at 3 gear the rpm was 14667, from 3 to 4 the rpm was 13704, from 4 to 5 the rpm was 13932',
  'Ideal Lap Gear Change RPMs': 'at 3 gear the rpm was 14622, from 3 to 4 the rpm was 13998, from 4 to 5 the rpm was 14490'},
 {'Segment': 'T11',
  'Fast Lap Gears used': [2, 3],
  'Ideal Lap Gears used': [2, 3],
  'Fast Lap Max RPM': '14892 rpm',
  'Ideal Lap Max RPM': '14826 rpm',
  'Fast Lap Min RPM': '8950 rpm',
  'Ideal Lap Min RPM': '9222 rpm',
  'Fast Lap Gear Change RPMs': 'at 2 gear the rpm was 8950, from 2 to 3 the rpm was 10146',
  'Ideal Lap Gear Change RPMs': 'at 2 gear the rpm was 9222, from 2 to 3 the rpm was 10014'},
 {'Segment': 'T10',
  'Fast Lap Gears used': [2, 3],
  'Ideal Lap Gears used': [2, 3],
  'Fast Lap Max RPM': '13698 rpm',
  'Ideal Lap Max RPM': '13266 rpm',
  'Fast Lap Min RPM': '7104 rpm',
  'Ideal Lap Min RPM': '7092 rpm',
  'Fast Lap Gear Change RPMs': 'at 3 gear the rpm was 13147, from 3 to 2 the rpm was 9708',
  'Ideal Lap Gear Change RPMs': 'at 3 gear the rpm was 12454, from 3 to 2 the rpm was 11160'}]
```

Figure 24: ML model's analysis overview of gear channel generated from randomly selected unseen data.

After identifying the rider's least efficient sector, the ML model investigated variations between channels to determine why the time discrepancy existed. The model cataloged key metrics pertaining to each channel's performance during the ideal and fast lap. Ultimately, it generated a concise text summary, which the chatbot could seamlessly interpret and convey. As an example, Figure 24 shows the gear channel's performance metrics for the track's final sector. Achieving optimal gearing performance requires not only selecting the appropriate gear promptly, but also monitoring the engine speed in order to maintain it within the power band on a consistent basis. Fundamentally, each channel has distinct priority areas that ensure the rider can maximize the performance of the motorcycle (Segers, 2014).

A comprehensive and nuanced evaluation of rider performance can only be achieved by equipping the AI model with extensive contextual information. Likewise, humans require sufficient

information to reach a firm conclusion. For this purpose, two OpenAI models were strategically deployed in sequence. The initial model analyzed time variations between the ideal and fast lap, paving the way for in-depth evaluation and specific feedback. Once key areas were spotlighted, the secondary model delved into the rider's metrics, scrutinizing aspects like braking efficiency, throttle modulation, gear shifts and beyond.

```
Based on the data, we can see that the most significant time differences occur in sectors 3 and 4, where you are losing 0.075 seconds and 0.195 seconds respectively. This suggests that these sectors should be your primary focus for improvement.
```

```
In terms of specific segments, T12 and T11 are areas where you're losing time. In T12, you're slower by 0.175 seconds, and in T11, you're slower by 0.05 seconds. Interestingly, you're faster in T10 by 0.035 seconds, which shows that you have the potential to improve in other areas as well.
```

```
Looking at the speed variations, in T12, your average speed is lower by 2.94 km/h, and your top speed is lower by 5.20 km/h. This suggests that you could potentially carry more speed through this segment. In T11, your average speed is slightly higher, but your top speed is lower, indicating that you might be braking too early or not accelerating hard enough out of the corner.
```

```
In T10, where you're faster, your average speed is slightly lower, but your top speed is higher. This suggests that you're carrying more speed into the segment, which is helping you gain time.
```

```
Now, let's dive deeper into the data using metrics like brake, throttle, gears, front and rear suspension, and wheel slip to understand and enhance the rider's performance further.
```

Figure 25: GPT-3.5-turbo first model session overview.

In this experiment, both models were powered by GPT-3.5-turbo, which was selected because it was the most advanced model available free of charge. According to Figure 19, the initial AI model was capable of identifying sectors that merited further investigation. Moreover, the bot systematically segmented the sectors in an effort to provide a simple analysis of what was contributing to the slower lap times observed.

```
Based on the provided data, here are some detailed observations and suggestions:
```

```
1. Braking: In T10, you are braking for a shorter duration and distance than in the ideal lap, but with higher brake pressure. This could be causing you to lose speed and time. Try to brake over a longer distance with less pressure to maintain a higher speed.
```

```
2. Throttle: In T12, you are on the throttle for a longer duration and distance than in the ideal lap. This could be causing you to lose time. Try to be more efficient with your throttle application to maintain a higher average speed.
```

```
3. Gear and RPM: In T12, your gear changes are happening at lower RPMs than in the ideal lap. This could be causing you to lose speed and time. Try to shift gears at higher RPMs to maintain a higher speed.
```

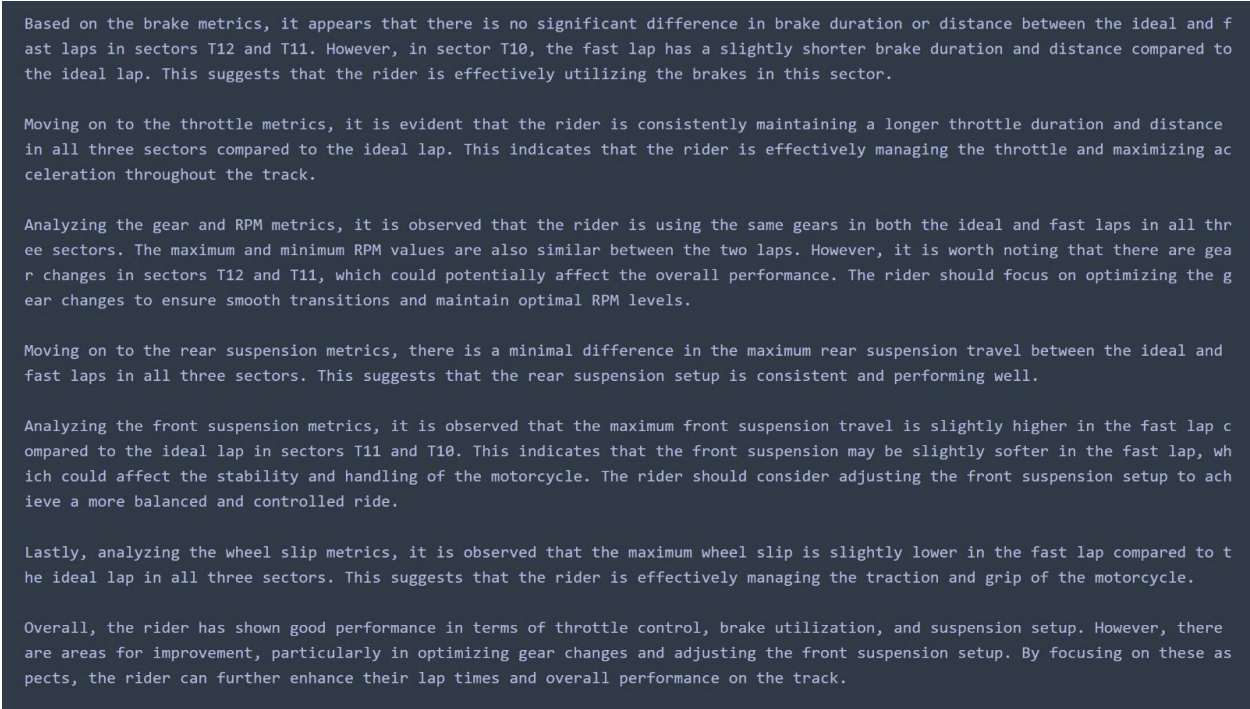
```
4. Suspension: In T11 and T10, your rear and front suspension are traveling more than in the ideal lap. This could be causing instability and loss of speed. Try to adjust your suspension settings to reduce travel and increase stability.
```

```
5. Wheel Slip: In T11 and T10, your wheel slip is higher than in the ideal lap. This could be causing you to lose traction and speed. Try to manage your throttle application and braking to reduce wheel slip and maintain a higher speed.
```

```
Remember, the key to a fast lap is not just about top speed, but maintaining a high average speed throughout the lap. This often means braking later, accelerating earlier, and carrying more speed through the corners.
```

Figure 26: GPT-3.5-turbo second model performance analysis.

Figure 26 showcases the final output generated by the second AI model. Essentially, this represents what the end user would see. While the AI model excelled at converting ML outputs into coherent sentences, it fell short in terms of analytical accuracy and the quality of its recommendations. A case in point is the brake pressure assessment, where it inaccurately advised the rider to apply less brake pressure over an extended duration to retain higher speeds for the upcoming turn, contrary to the optimal strategy. Before making any changes to the ML model to potentially address these shortcomings, it was considered to first explore the premium OpenAI model version, GPT-4 (OpenAI, 2023).

A screenshot of a text-based analysis generated by the GPT-4 model. The text is displayed on a dark blue background with white monospaced font. It provides a detailed comparison of rider performance across various metrics (brake, throttle, gear, suspension, wheel slip) for ideal, fast, and predicted laps across three sectors (T10, T11, T12).

Based on the brake metrics, it appears that there is no significant difference in brake duration or distance between the ideal and fast laps in sectors T12 and T11. However, in sector T10, the fast lap has a slightly shorter brake duration and distance compared to the ideal lap. This suggests that the rider is effectively utilizing the brakes in this sector.

Moving on to the throttle metrics, it is evident that the rider is consistently maintaining a longer throttle duration and distance in all three sectors compared to the ideal lap. This indicates that the rider is effectively managing the throttle and maximizing acceleration throughout the track.

Analyzing the gear and RPM metrics, it is observed that the rider is using the same gears in both the ideal and fast laps in all three sectors. The maximum and minimum RPM values are also similar between the two laps. However, it is worth noting that there are gear changes in sectors T12 and T11, which could potentially affect the overall performance. The rider should focus on optimizing the gear changes to ensure smooth transitions and maintain optimal RPM levels.

Moving on to the rear suspension metrics, there is a minimal difference in the maximum rear suspension travel between the ideal and fast laps in all three sectors. This suggests that the rear suspension setup is consistent and performing well.

Analyzing the front suspension metrics, it is observed that the maximum front suspension travel is slightly higher in the fast lap compared to the ideal lap in sectors T11 and T10. This indicates that the front suspension may be slightly softer in the fast lap, which could affect the stability and handling of the motorcycle. The rider should consider adjusting the front suspension setup to achieve a more balanced and controlled ride.

Lastly, analyzing the wheel slip metrics, it is observed that the maximum wheel slip is slightly lower in the fast lap compared to the ideal lap in all three sectors. This suggests that the rider is effectively managing the traction and grip of the motorcycle.

Overall, the rider has shown good performance in terms of throttle control, brake utilization, and suspension setup. However, there are areas for improvement, particularly in optimizing gear changes and adjusting the front suspension setup. By focusing on these aspects, the rider can further enhance their lap times and overall performance on the track.

Figure 27: GPT-4 model performance analysis.

Rather than directly suggesting a solution that may be subject to error, the model meticulously analyzed the problem, highlighting potential areas of concern and reasons behind them. The primary objective was to provide engineers with a deeper understanding of the topic, rather than to suggest a solution directly. Given its efficacy in fulfilling this aim, the GPT-4 model was employed for the remainder of the project. Following a comprehensive comparison between the ideal and fast lap, the same rigorous process was applied to compare the ideal lap with the predicted lap. Therefore, identify further opportunities for improving performance.

Prompt engineering involves the careful design, enhancement and fine-tuning of prompts presented to a ML model to yield precise, relevant and context-driven results. Notably, different iterations of GPT-3.5-turbo and GPT-4 demonstrated varied interpretations of problems and solutions presented to users. An inconsistency that is less than ideal in high-stress fields like data engineering. To address these variations, it was crucial to establish a clear role, operational sequence and content specifications for the AI model prior to any analytical assignments. Several prompt configurations were tested and evaluated. The following prompts have been identified for the first and second bots:

“You are a data engineer working with a motorcycle rider. Your task is to interpret and understand the telemetry data from two laps: the “ideal lap” (a theoretical best lap from combined best sectors) and the “fast lap” (the rider's fastest real lap). When providing actionable insights and suggestions, it is crucial to reference the specific numbers and details in the data. This will help the rider understand the exact differences and areas for improvement. Use only the data provided and do not make up any details.”

“You are a data engineer working with a motorcycle rider. You are about to receive a summary of the most recently analyzed telemetry session by another motorcycle data engineer. Your task is to dive deeper into the telemetry metrics such as brake, throttle, gears, front and rear suspension, and wheel slip. These metrics will be compared between the “ideal lap” and the “fast lap”. It is imperative to reference the specific numbers and details from the data to help the rider understand the precise differences and areas for improvement. Use only the data provided and do not make up any details.”

A consistent prompt structure was used for comparing the ideal lap with the predicted lap. Recognizing the importance of user accessibility and understanding, efforts have been directed towards integrating visual graph comparisons with textual explanations. The result was the design of a user interface (UI), also known as the front-end (Plotly, 2023). The user only needs to enter the desired session for analysis into the front-end, the back-end would handle all data processing activities and the final results would be relayed to the front-end for review.

4 Results

Once the telemetry prediction ML model was trained, it was saved as a pickle file, which enabled immediate access with minimal computing resources. As highlighted earlier, a subset of the test data was set aside for validation due to its accompanying setting sheets. Free practice 3 (FP3), which recorded the best lap time of 94.00 seconds, served as the benchmark for the study. Appendix G contains additional results for the subsequent qualifying session.

4.1 Fast Against Ideal Lap

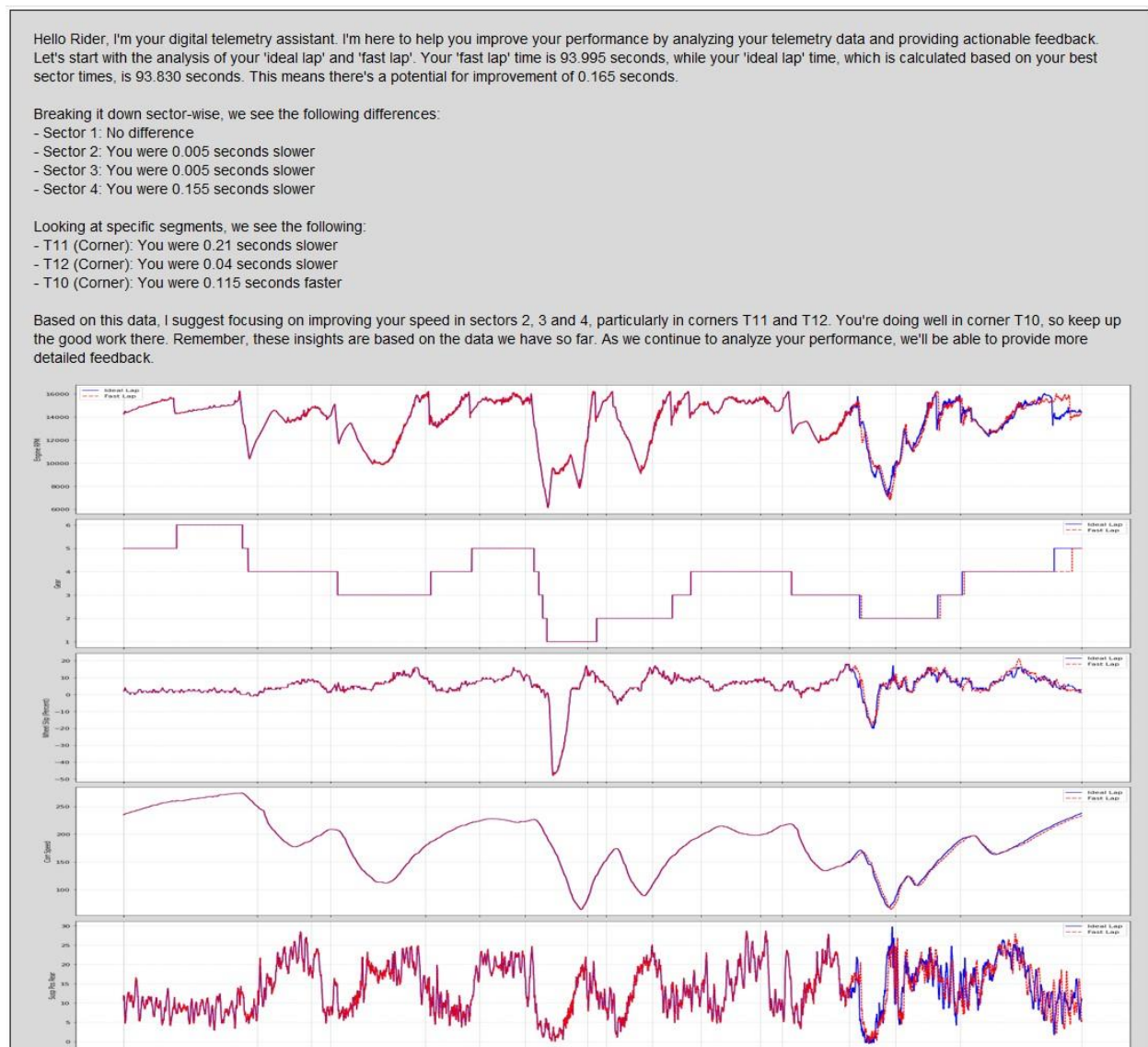


Figure 28: 1st part of ideal against fast lap AI analysis from FP3 session.

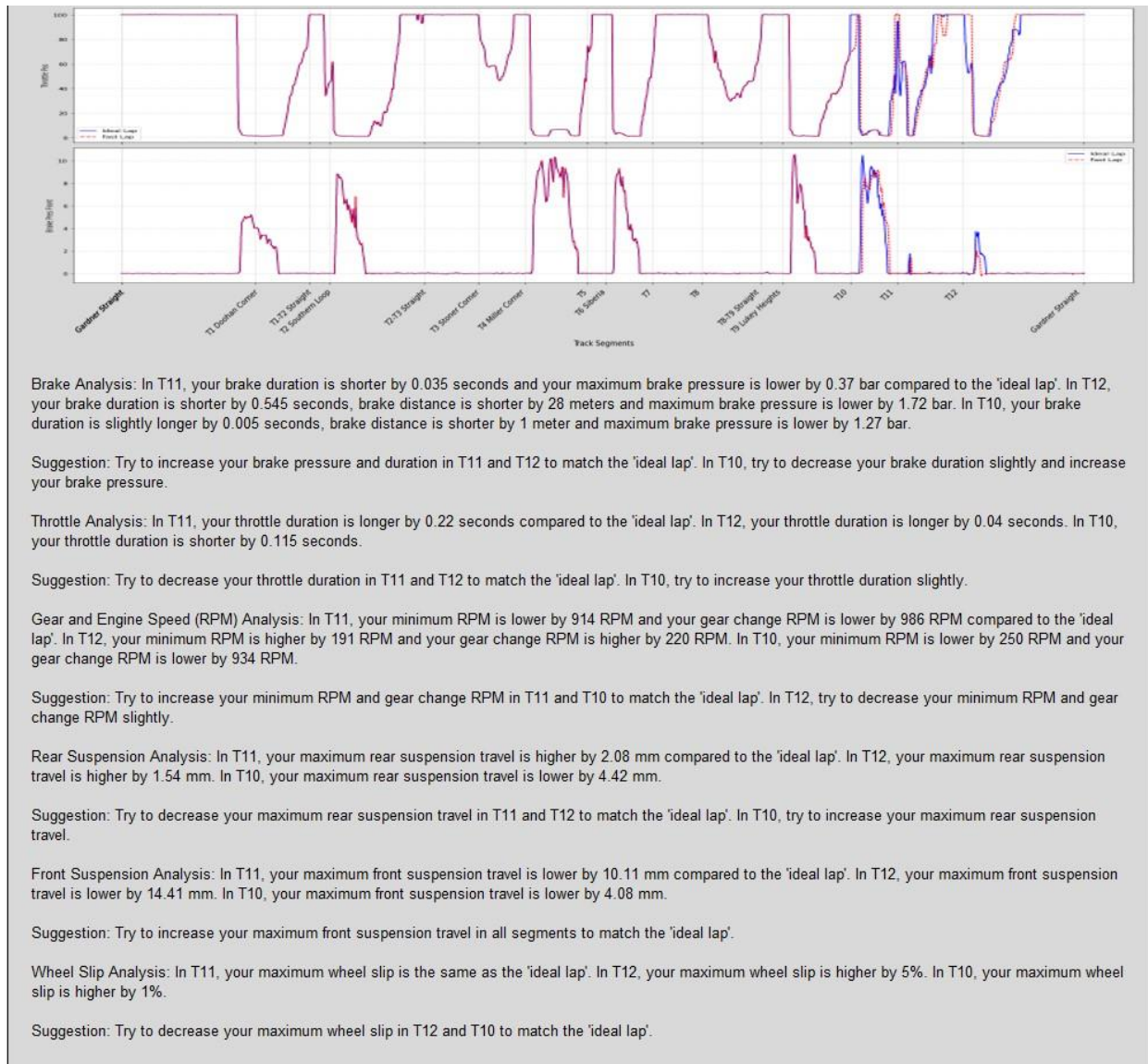


Figure 29: 2nd part of ideal against fast lap AI analysis from FP3 session.

4.2 Ideal Against Predicted Lap

Let's start again with the overall comparison. Your 'ideal lap' time is currently 93.830 seconds, while the 'lap record' stands at 93.072 seconds. This means there's a potential for improvement of 0.758 seconds.

Now, let's break it down sector by sector:

- Sector 1: Your time is 0.210 seconds slower than the lap record.
- Sector 2: You're lagging by 0.197 seconds.
- Sector 3: The gap is 0.147 seconds.
- Sector 4: You're behind by 0.204 seconds.

This data shows that your performance in Sector 1 needs the most improvement. Let's further break down Sector 1 into its segments:

- Gardner Straight: You're 0.070 seconds slower.
- T1 Doohan Corner: You're behind by 0.037 seconds.
- T1-T2 Straight: The gap is 0.013 seconds.

From this, we can see that the Gardner Straight is the segment where you lose the most time.

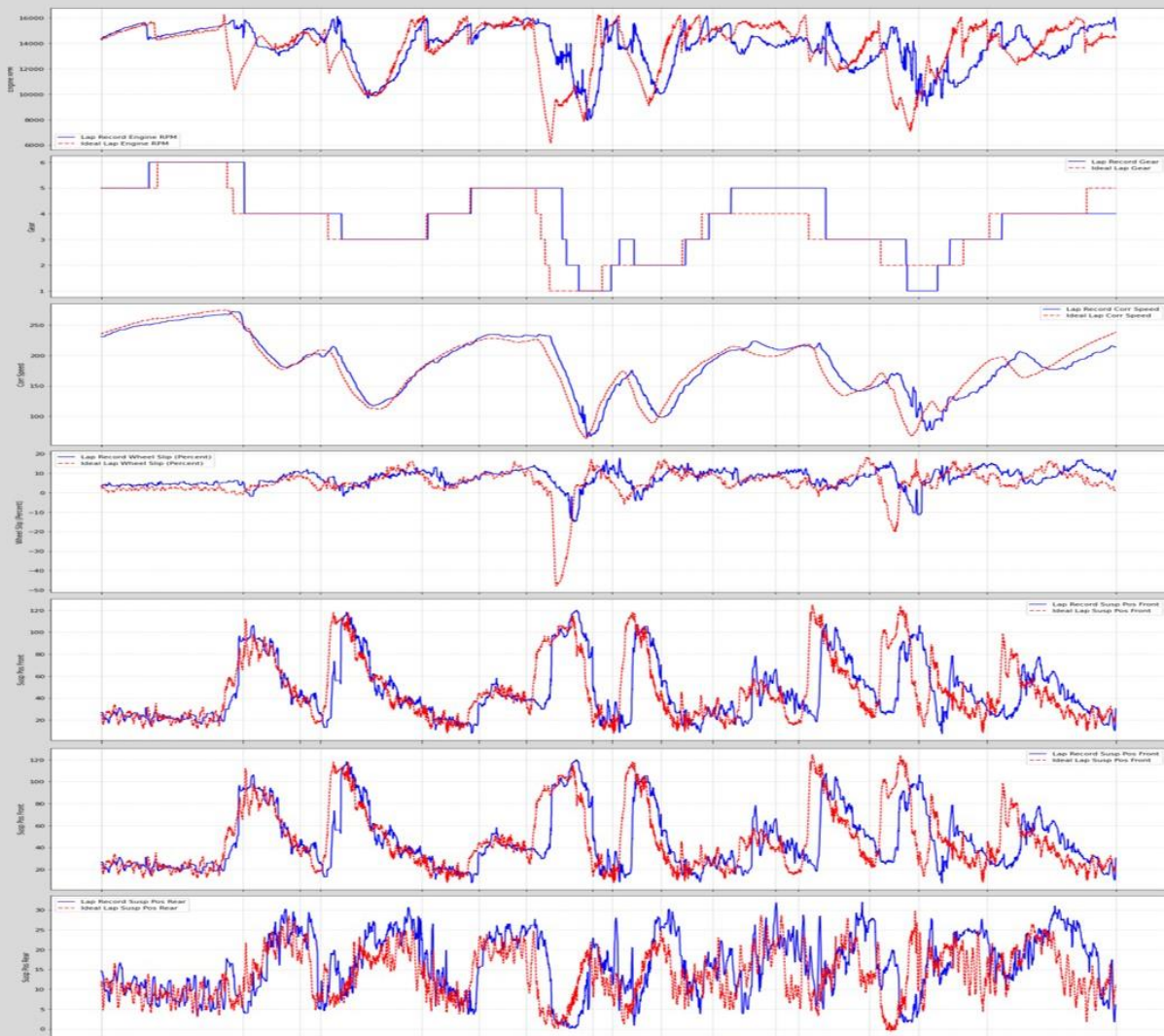


Figure 30: 1st part of ideal against predicted lap AI analysis from FP3 session.

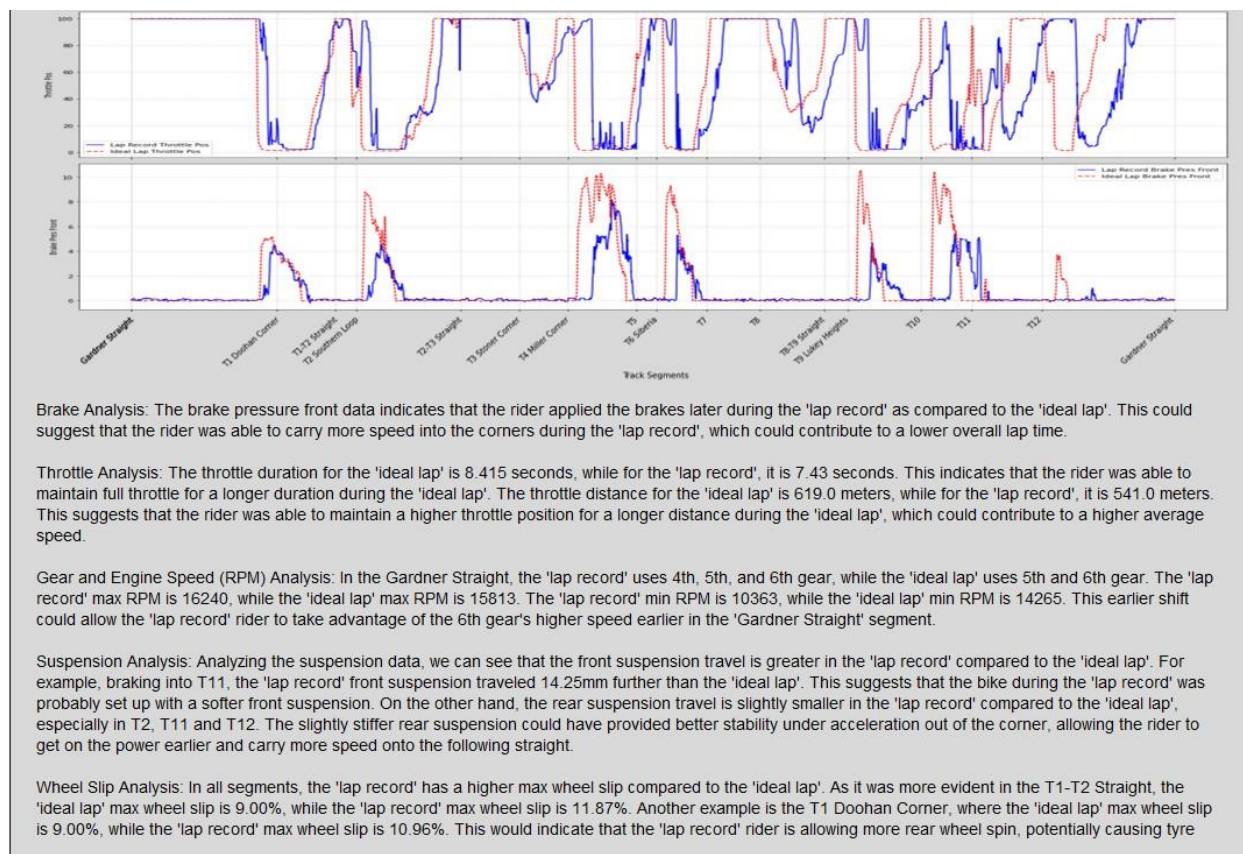


Figure 31: 2nd part of ideal against predicted lap AI analysis from FP3 session.

5 Discussion

Telemetry assistant's results were based on a typical chart for analyzing motorcycle performance factors, similar to MoTeC software shown in Figure 14, Section 3.2. In effect, two parallel analyses of the data were conducted. Initial comparisons between the fastest lap time (represented in red) and the ideal lap time (represented in blue), as illustrated in Figures 28 and 29. Afterward, the ideal lap (now displayed in red) was compared with the predicted lap record (blue) in Figures 30 and 31. To enrich the UI, a third AI model was incorporated. Following evaluations by the previous OpenAI entities, the concluding GPT-4 model provided an initial performance summary comparable to a split-time report. It then proceeded to dissect the telemetry data in greater detail, proposing optimization measures.

5.1 Rider Performance Data Analysis

At the end of the selected FP3 session, the team's crew chief documented the following debrief: "T2 exit, fourth gear loses drive. T9, can't spin the rear tire, not enough power. Bit richer? Off brake T6, T9, T11 feels chatter but when crack on the gas it ploughs. T2 check TC, needs a bit more." (Intermoto Ponyexpres, 2015). These comments indicate that most of the rider's difficulties were encountered in sector 4. Figure 28, which compares the ideal lap time to the fastest lap time, clearly demonstrates the rider's efforts to optimize their trajectory in the fourth sector. Moreover, the AI assistant identified T10 as the most challenging corner for the rider. Considering that a compromised exit from T9 would naturally affect T10, this observation aligned with the rider's feedback.

The reliability of the telemetry assistant's analysis was confirmed by an interview with Andrew Harrison, a seasoned motorcycle data engineer. As part of the first phase of the analysis, the following feedback was received: "Typically, the rider reviews the split time report independently. If they notice an improvement in a particular sector compared to their fast lap, we collaborate to identify the cause of the performance difference. Considering such scenarios, the AI model's analytical capabilities can be of great value. Its recommendations, however, raise some concerns. The majority of its observations are accurate, but some of them are deficient in critical details that could result in significant problems. As an example, the model advised longer and harder braking before T12. Yet, it omitted to specify that the entry speed into T12 should also be reduced. A rider

can optimize the T12 exit by braking harder and longer, gaining speed for the upcoming straight, and thus improving overall lap time."

5.2 Further Improvement Data Analysis

The second analysis, presented in Figures 30 and 31, revealed a noticeable offset between the real and predicted values. Later in the development process, MoTeC was understood to calculate its traveled distance using the corrected speed, which is based on the front and rear wheel speed channels. Consequently, when the rider encounters wheel slip, the accuracy of this channel can be slightly compromised, affecting the reported distance. A standard practice, as explained by Trapani (2021), would be to match the circuit bumps by looking at the front and rear suspension and then analyze corner by corner. Thus, the AI's results were still deemed successful, but their evaluation was challenged by the lack of synchronization options.

Overall, the predicted telemetry data generally matched the expected performance patterns. Filtered engine speed, correlated speed, and gear channels exhibited a more realistic appearance. However, the engine speed channel presented noticeable inconsistencies. For instance, the ML model predicted that the rider would maintain the 2nd gear after exiting T6. Contrary, a sudden decrease followed by a sharp increase in engine speed indicated a gear change. The engine speed was increased by approximately 4000 rpm between T11 and T12 even though the gearing was the same between the two times. This disparity seemed unlikely given the acceleration window available.

Interestingly, the model obtained a different gearing strategy in T11 by using 3rd. In FP2, the session prior to the chosen one, the following comment was made during the debrief: "Check T11 2nd or 3rd? On data no difference in time but the bike is more settled using 3rd. More grip allowed better shifting T12. Wants more engine brake into T10, needs more stop on last part of entry." (Intermoto Ponyexpres, 2015). Sector 4 was identified by the model as the second least efficient sector, resulting in a different approach. This might explain the preference for third gear, provided that sufficient speed is available within this segment.

As a final note, Andrew Harrison evaluated the second phase of the data analysis and provided the following feedback: "Once again, the AI assistant did a great job analyzing the data. Providing an overview of channel performance was a positive touch since all sectors have potential for improvement. Our post-session protocol usually involves scanning channels' minimum values to

detect anomalies due to sensor miscalibration. Following that, we review the maximum and average values to determine the primary areas of focus... Despite the AI assistant's efforts, it still lacks specificity in its recommendations. For instance, it suggested increasing wheel slip in the first sector and maintaining the throttle open for an extended duration. Although, it was unclear whether the suggestion was to sustain throttle engagement from the same pickup point for a longer period of time or to engage the throttle earlier in the sequence to achieve a higher average wheel slip. In its current form, the telemetry assistant software could serve as a valuable asset for amateur teams and riders, guiding them through their data analysis process. However, this product requires further refinement in order for it to be included in top-tier professional racing."

6 Conclusions

The primary objective of this master's project was to develop and design innovative telemetry analysis software. This was achieved by harnessing ML capabilities to predict racing performance across various channels using only a desired lap time. Leveraging OpenAI models, the disparities between predicted and actual telemetry data were translated into clear and insightful analyses. Ultimately, this software package was envisioned to assist engineers in analyzing motorcycle telemetry data, streamlining their efforts to achieve optimal lap times.

6.1 Findings

Insights from comprehensive literature reviews paved the way for the development of a many-to-one ML model. Specifically, a Random Forest Regression model was trained using data from 490 laps on the Phillip Island circuit obtained from a SuperSport bike. Upon testing this model with 120 unseen laps, it achieved an impressive average R^2 score of 0.948. Essentially, the ML models could predict lap times within a mere 0.05 seconds using only performance channels, such as correlated speed, throttle input, brake pressure, etc.

Reversing this process to forecast channel performance based on lap time using the same model configuration proved challenging. Initially, incorporating the Polynomial Regression model into the RF framework seemed promising. However, the high coefficients required to align with the non-linear channel relationships complicated the convergence process. As a remedy, various feature engineering techniques, including track segmentation, sessions and lap markers were integrated. The subsequent performance of the ML model exhibited marked improvements, reflected in an average MSE and MAE score of 0.03 for unseen data. An exception was the engine speed channel, characterized by its inherent noise. After augmenting the feature engineering set with RFEs and re-training a secondary model designed specifically for the engine speed channel, the MSE and MAE achieved a remarkable improvement of over 1000%.

In the quest to pinpoint areas of enhancement, the ML model was designed to monitor the time variance between the fastest, ideal and predicted lap times. As the time variance increased, differences in channel performance were documented. The task of converting these intricate data summaries into meaningful, coherent analyses was then assigned to two OpenAI models. The first model provided an overview of the analyzed session. By receiving additional contextual information from "another motorcycle engineer", the following AI model was able to gain a deeper

understanding of the data. Initial attempts with GPT-3.5-turbo lacked precision, prompting a shift to GPT-4. Through tailored prompt engineering, the model was further refined to exhibit professional behavior, correct terminology and a consistent structure.

Finally, a chatbot interface was designed, integrating explanations from the GPT-4 models with visual insights from the ML algorithm. The final product was validated over a full race weekend supported by setting sheets. The chatbot successfully pinpointed areas of difficulty that the rider genuinely encountered throughout the event. Moreover, it highlighted specific performance channels for extra attention, of which 80% matched the documented solutions. In addition, a motorcycle data engineer with over two decades' experience evaluated the novel platform. Among its advantages, he cited its user-friendly interface, swift insight delivery and vivid data visualization. However, he also noted a tendency for the chatbot to occasionally overwhelm users with excessive or unclear suggestions, leading to confusion. Despite this, the telemetry assistance package was considered as an indispensable asset for any racing team.

6.2 Future Work

A primary area for future exploration is the telemetry prediction ML model. While it produced satisfactory outcomes for most performance parameters, notable noise remained even after filtering functions were applied, suggesting decisional uncertainty. It was discovered that the ML model increased its predictive capabilities by around 1% for every 25 laps added to the training dataset. This observation raised the prospect of untapped potential within the ML model through further training with an expanded dataset. Furthermore, the hypothesis emerged that broadening the range of performance channels might empower the model to discern more complex data correlations and enhance its overall reliability.

Further research will also be required on the OpenAI model. While the GPT-4 chatbot demonstrated strength in data analysis, it sometimes fell short in articulating the basis for its results, leading to poor suggestions. Supervised training using domain-specific literature could offer a viable strategy to better adapt the model to motorcycle telemetry applications. Furthermore, input from experienced motorcycle data engineers and racers could provide pivotal insights to refine the chatbot's output quality (OpenAI, 2023).

During the validation phase, it also became evident that a vital aspect of data engineering was the assessment of bike sensors' health to ensure reliable data acquisition. In this particular project, this task was conducted manually. Thus, engineers would need to first examine the data, export it and upload it to telemetry assistant software. The use of such an approach is not best suited to real-world applications, since it disturbs workflow and diminishes the product's value. Ideally, the telemetry assistant should seamlessly interface with the telemetry acquisition software, enabling real-time data reliability assessment. Further, by training the ML model on health channels, faster issue detection becomes feasible. This integration also empowers the model to identify deeper channel interdependencies, potentially enhancing its predictions for performance channels.

The scope of this study did not fully explore the flexibility of the product developed. As a result, a comprehensive assessment should involve evaluations by racing teams at different levels, across various tracks and motorcycle models. Moreover, the ML model utilized statistical tree derivations to discern relationships between channels. In theory, this model can be extended to the automotive sector with appropriate training on relevant telemetry data, offering considerable opportunities for expansion. This potential is demonstrated by the fact that motorsport vehicles, like those in NASCAR, can produce data for more than 1,000 laps over a single race weekend.

6.3 Reflection

The project achieved its core objectives and, in many ways, exceeded expectations given its pioneering nature. Despite challenges and constraints, the research demonstrated the immense potential of combining ML, AI and telemetry data within the motorcycle industry. This endeavor has undoubtedly laid the groundwork for future research in this domain.

A key contributor to the project's success was a risk assessment conducted from the very beginning. This process identifies potential hazards, evaluates their probability of occurring, assesses their severity if they occur and plans mitigation measures. For instance, during the ML modelling process, up to 85 GB of random access memory (RAM) was required and a complete hyperparameter tuning process could take up to 12 hours. Therefore, a transition to Google Colab was deemed necessary, as outlined in Table 6. A prominent cloud-based platform that offers not only the advantage of tensor processing units (TPUs), which is crucial for speeding up ML tasks, but also the capability to manage large amounts of memory (Colab, 2023).

Table 11: Risk assessment.

Problem	Likelihood	Severity	Strategy
Data quality.	Low	High	Perform a manual review to ensure that all performance channels across all datasets are consistent and of high quality.
Data quantity.	Medium	High	Investigate techniques for synthetically expanding an existing dataset to enhance its volume.
Difficulties with Python.	Low	Medium	Take online Python courses or refer to Python programming books for a better understanding and troubleshooting of the language.
Difficulties with ML modelling.	Low	High	Adapt regularization techniques to counter potential overfitting or underfitting issues when developing the ML model.
Inappropriate evaluation metrics.	Medium	High	Conduct a thorough literature review to identify the most appropriate evaluation metrics aligned with the project objectives.
Computational resource limitations.	Medium	High	Consider acquiring additional computational resources, optimizing code for performance or upgrading to cloud-based services.
Difficulties with OpenAI integration.	Medium	High	Consider consulting OpenAI's official documentation and seeking guidance from AI experts.
OpenAI misinterpretation of results.	Low	High	Provide additional validation checks, domain-specific constraints or user feedback loops to correct interpretations.

Segmenting the project into distinct tasks and milestones was also essential to its successful execution. Regular monitoring of progress offered a clear overview and allowed for flexibility in strategy adaptation. For example, the telemetry ML prediction model encountered consistent challenges with the engine speed channel, leading to deviations from the original timeline provided in Appendix A. As a result, an alternative approach was adopted, as described in Section 3.4.2 and a revised Gantt chart shown in Table 7 was constructed.

Table 12: Updated Gantt Chart.

NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION in days	% OF TASK COMPLETE	MARCH							APRIL				MAY				JUNE				JULY				AUGUST				SEPTEMBER				
						6	13	20	27	3	10	17	24	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25			
1	Research																																				
1.1	Literature Review	06/03/23	23/04/23	42	100%																																
1.2	Risk Assessment	24/04/23	01/05/23	7	100%																																
2	Data Acquisition																																				
2.1	Data Collection	02/05/23	23/05/23	21	100%																																
2.2	Analysis + Extraction	24/05/23	31/05/23	7	100%																																
2.3	Preprocessing	01/06/23	22/06/23	21	100%																																
3	Modeling																																				
3.1	Feature Engineering	23/06/23	07/07/23	14	100%																																
3.2	Lap Time ML Predictor	23/06/23	07/07/23	14	100%																																
3.3	Telemetry ML Predictor	08/07/23	05/08/23	28	100%																																
3.5	OpenAI Chatbot	06/08/23	13/08/23	7	100%																																
3.6	User Interface	14/08/23	21/08/23	7	100%																																
4	Evaluation																																				
4.1	ML Model Training and Testing	03/07/23	21/08/23	49	100%																																
4.2	Performance Analysis	22/08/23	25/08/23	3	100%																																
4.3	Product Validation	26/08/23	29/08/23	3	100%																																
5	Report Writing																																				
5.1	Report Drafting	30/08/23	02/09/23	3	100%																																
5.2	Report Writing + Referencing	03/09/23	08/09/23	5	100%																																
5.3	Reprot Proofreading + Submission	09/09/23	10/09/23	1	100%																																
5.4	Poster + Viva	11/09/23	19/09/23	8	100%																																

References

- Aggarwal, C., 2018. Neural Networks and Deep Learning. New York: Springer.
- Bergstra, J. & Bengio, Y., 2012. Random search for hyper-parameter optimization. The Journal of Machine Learning Research, Volume 13, pp. 281-305.
- Biniewicz, J. & Pyrz, M., 2023. A quasi-steady-state minimum lap time simulation of race motorcycles using experimental data. International Journal of Vehicle Mechanics and Mobility.
- Bradley, J., 2018. The Racing Motorcycle: Volume 3: An Introduction to Chassis Set Up. 3 ed. s.l.:John Bradley.
- Brown, T., Mann, B., Ryder, N. & Subbiah, M., 2020. Language Models are Few-Shot Learners, s.l.: Johns Hopkins University.
- Colab, 2023. Machine learning. [Online]
Available at: <https://colab.research.google.com/#scrollTo=OwuxHmxllTwN>
- Fan, C., Chen, M., Wang, X. & Huang, B., 2021. A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. Frontiers in Energy Research, Volume 9, pp. 1-17.
- FIM, 2015. FIM SUPERBIKE, SUPERSPORT & SUPERSPORT 300 WORLD CHAMPIONSHIPS REGULATIONS, s.l.: FIM.
- FIM, 2015. Phillip Island, 20-21-22 February 2015 Supersport - Results Race. [Online]
Available at:
https://resources.worldsbk.com/files/results/2015/AUS/SSP/001/CLA/Results.pdf?version=3f254ea62985e67d70d5751fd79112ed&_ga=2.10760627.66153007.1693405984-1879803126.1691932994
- Georgiev, I., 2022. Motorcycle Chassis Modelling, Performance Predictions and Race Weekend Efficiency, Swansea: UWTSD.
- Georgiev, I., 2023. An Investigation of the Relationship between Chassis Settings and Optimal Trajectory for Improving the Lap-Time of a Racing Motorcycle, Swansea: UWTSD.

Gheisari, M., Wang, G. & Bhuiyan, M. Z. A., 2017. A Survey on Deep Learning in Big Data. 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 173-180.

Gupta, A., 2021. XGBoost versus Random Forest. [Online]
Available at: <https://medium.com/geekculture/xgboost-versus-random-forest-898e42870f30>

Hastie, T., Tibshirani, R. & Friedman, J., 2001. The Elements of Statistical Learning. 2nd Edition ed. s.l.:Springer.

Hojaji, F., Toth, A. & Campbell, M., 2023. A Machine Learning Approach for Modeling and Analyzing of Driver Performance in Simulated Racing, Limerick,: University of Limerick.

Ibbot, A., 2021. MotoGP Performance Riding Techniques. Andy Ibbot ed. s.l.:s.n.

Intermoto Ponyexpres, 2015. ZX6R Phillip Island, s.l.: UWTSD.

Kuhn, M. & Johnson, K., 2019. Feature Engineering and Selection: A Practical Approach for Predictive Models. s.l.:CRC Press.

Massegur, D., 2022. How Artificial Intelligence Will Shape The Fastest Race Car. [Online]
Available at: <https://motorsportengineer.net/how-artificial-intelligence-will-shape-the-fastest-race-car/>

Nguyen, L., 2017. A Vehicle Classification Algorithm based on Telematics Data, s.l.: Massachusetts Institute of Technology.

NumFOCUS, 2023. User Guide. [Online]
Available at: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

NumPy, 2023. numpy.polyfit. [Online]
Available at: <https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>

OpenAI, 2023. Finr-tunning. <https://platform.openai.com/docs/guides/fine-tuning> ed. s.l.:s.n.

OpenAI, 2023. Models. [Online]
Available at: <https://platform.openai.com/docs/models/overview>

Peters, J. & Gerdes, J., 2021. Machine Learning for Modeling and Analyzing of Race Car Drivers, s.l.: Technische Universität.

Phillip Island Circuit, 2020. Motorcycle Lap Records. [Online]
Available at: <https://www.phillipislandcircuit.com.au/wp-content/uploads/2021/01/Lap-Record.pdf>

Phillip Island Circuit, 2023. About. [Online]
Available at: <https://www.phillipislandcircuit.com.au/circuit-info/about/>

Plotly, 2023. Overview of Dash & Dash Apps. <https://plotly.com/dash/> ed. s.l.:s.n.

Pontin, S., 2023. AIBased Race Strategy Assistant and Car data Monitor, s.l.: Luleå University of Technology.

Python, 2023. Python object serialization. [Online]
Available at: <https://docs.python.org/3/library/pickle.html>

Remonda, A. et al., 2021. Formula RL: Deep Reinforcement Learning for Autonomous Racing using Telemetry Data, Graz: Know-Center.

Sarker, I., 2021. Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. SN Computer Science, pp. 1-22.

Schick, T. & Schutze, H., 2021. True Few-Shot Learning with Prompts—A Real-World Perspective, Munich: Center for Information and Language Processing (CIS).

Schleinitz, J. v. et al., 2022. Race Driver Evaluation at a Driving Simulator using a physical Model and a Machine Learning Approach, s.l.: Cornell University .

Scikit-learn, 2023. OneHotEncoder. [Online]
Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>

Scikit-learn, 2023. Preprocessing data. [Online]
Available at: <https://scikit-learn.org/stable/modules/preprocessing.html>

Scikit-learn, 2023. sklearn.ensemble.RandomForestRegressor. [Online]
Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

Segers, J., 2014. Analysis Techniques for Racecar Data Acquisition. Second ed. Warrendale: SAE.

Sipola, C., 2017. Summarizing electricity usage, Edinburgh: University of Edinburgh.

- Trapani, M., 2021. A practical guide to race motorbike electronics. s.l.:Michele Trapani.
- VanderPlas, J., 2016. Python Data Science Handbook. s.l.:O'Reilly Media.
- Versloot, C., 2019. Amplifying the analyst: Machine Learning approaches for buried utility characterization , s.l.: University of Twente.
- Wu, Y., Bamman, D. & Russell, S., 2018. Adversarial Training for Relation Extraction. Conference on Empirical Methods in Natural Language Processing, p. 1778–1783.

7 Appendices

The following appendices provide supplementary material related to the research conducted in this master's project. Their purpose is to provide detailed information for a deeper understanding of the research process and to promote transparency and reproducibility.

7.1 Appendix A

Table A-1: Initial Gantt Chart

NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION in days	% OF TASK COMPLETE	MARCH				APRIL				MAY				JUNE				JULY				AUGUST				SEPTEMBER			
						6	13	20	27	3	10	17	24	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18
1	Research																																
1.1	Literature Review	06/03/23	23/04/23	42	100%																												
1.2	Risk Assessment	24/04/23	01/05/23	7	100%																												
2	Data Acquisition																																
2.1	Data Collection	02/05/23	23/05/23	21	100%																												
2.2	Analysis + Extraction	24/05/23	31/05/23	7	100%																												
2.3	Preprocessing	01/06/23	15/06/23	14	100%																												
3	Modeling																																
3.1	Feature Engineering	16/06/23	23/06/23	7	80%																												
3.2	Lap Time ML Predictor	24/06/23	08/07/23	14	100%																												
3.3	Telemetry ML Predictor	09/07/23	23/07/23	14	50%																												
3.5	OpenAI Chatbot	24/07/23	31/07/23	7	0%																												
3.6	User Interface	01/08/23	08/08/23	7	0%																												
4	Evaluation																																
4.1	ML Model Training and Testing	03/07/23	14/08/23	42	20%																												
4.2	Performance Analysis	14/08/23	21/08/23	7	0%																												
4.3	Product Validation	21/08/23	24/08/23	3	0%																												
5	Report Writing																																
5.1	Report Drafting	25/08/23	30/08/23	5	10%																												
5.2	Report Writing + Referencing	31/08/23	06/09/23	6	0%																												
5.3	Reprot Proofreading + Submission	07/09/23	10/09/23	3	0%																												
5.4	Poster + Viva	11/09/23	18/09/23	7	0%																												

As previously mentioned, the hyperparameter tuning for each of the developed models took considerably longer than expected. Figure A-1 presents a Gantt chart drafted at the beginning of the project, which evidently did not allocate sufficient time for the modeling phase. This oversight resulted in certain areas, such as the engine speed channel predictions, not receiving the required attention, potentially affecting the project's quality.

7.2 Appendix B

The lap time prediction ML model was designed to forecast the fastest lap time within a session, based solely on performance channels. Optimal performance was achieved by setting "n_estimators" to 50, "max_depth" to undefined (None), "min_samples_leaf" to 2 and "min_samples_split" to 4. The model was tested in multiple unfamiliar sessions. Results were compared with those obtained from the MoTeC i2 Pro telemetry software, Table B-1.

Table B-1: MoTeC split times report for lap time prediction ML model validation.

Phillip Island, Race 1 - Phillip Island Patrick Jacobsen, PJ2015 2/20/2015, FP2 3:35:53 PM,					Eclectic	Rolling Minimum
	Lap 1	Lap 2	Lap 3	Lap 4		
1	26.179211	26.161578	26.127788	26.216472	26.127788	26.127788
2	24.793927	24.473896	24.690521	24.652653	24.473896	24.473896
3	18.523014	18.408958	18.493169	18.454983	18.408958	18.408958
4	25.503847	25.355569	25.488522	25.475891	25.355569	25.355569
Totals	95.000000	94.400000	94.800000	94.800000	94.366210	94.366210

```
# Load the new data
file_path = r"F:\Iriney\Master\2 AI\comparation Lap\GP_FP2_EX2_4Laps.csv"
new_data = pd.read_csv(file_path, skiprows=18, names=column_names)

# Split the data into separate laps
new_laps = split_lapsComp(file_path)

# Identify the fastest lap
lap_times = [lap_df['Time'].iloc[-1] for lap_df in new_laps]
fastest_lap_idx = lap_times.index(min(lap_times))
comparation_fastlap = new_laps[fastest_lap_idx]

# Prepare the feature data for the fastest lap
feature_values = [comparation_fastlap[feature].mean() for feature in features]

# Expand the features for the fastest lap
X_comparation_fastlap = expand_features(comparation_fastlap, features)

# Normalize the expanded features using the scaler from training
normalized_feature_values = scaler.transform(X_comparation_fastlap)

# Predict using the normalized features
predicted_lap_time_normalized = model.predict(normalized_feature_values)

# Print the predicted lap time
print("Predicted lap time:", predicted_lap_time_normalized[0])
```

Predicted lap time: 95.00754687070711

Figure B-1: Lap time prediction ML model validation example.

As an example, when presented with a session consisting of four complete laps, the difference in lap time was approximately 0.6 seconds, as shown in Figure B-1 and Table B-1. This was one of the less favorable validation outcomes. Nevertheless, the model demonstrated sufficient reliability for starting the development of the telemetry ML prediction model, as explained in Section 3.4.2.

7.3 Appendix C

At the early stages of the telemetry prediction model development, the track was segmented only into sectors. In the course of the project, more sophisticated feature engineering techniques were needed, resulting in a further subdivision of the track into straights and corners. Table C-1 shows the precise distances associated with each segment. Due to the absence of a publicly available track map, these segmentations were approximate.

Table C-1: Track segments.

Distance m	Event	Name	Sector No	Map
0	Straight	Gardner Straight	1	
620	Corner	T1 Doohan Corner	1	
870	Straight	T1-T2 Straight	1	
960	Corner	T2 Southern Loop	1	
1400	Straight	T2-T3 Straight	2	
1650	Corner	T3 Stoner Corner	2	
1860	Corner	T4 Miller Corner	2	
2150	Corner	T5	2	
2235	Corner	T6 Siberia	2	
2450	Corner	T7	3	
2675	Corner	T8	3	
2950	Straight	T8-T9 Straight	3	
3050	Corner	T9 Lukey Heights	3	
3360	Corner	T10	4	
3580	Corner	T11	4	
3875	Corner	T12	4	
4430	Straight	Gardner Straight	4	

The longitudinal and latitudinal coordinates of the Phillip Island circuit were sourced from Google Earth. Using these coordinates and the segments, a track map was constructed, as shown in Table C-1. This addition did not have any influence on the ML predicted results. However, it was considered a useful tool when analyzing telemetry data, so it was included in the UI.

Table C-2: Average prediction performance metrics comparison with additional segments.

Channel	MSE	MSE	MAE	MAE
	with Sectors	with Sectors & Events	with Sectors	with Sectors & Events
Engine Speed	538569.70	22906.46	469.55	279.69
Gear	0.11	0.07	0.17	0.07
Corr Speed	92.33	20.28	5.20	3.40
Wheel Slip	14.51	5.62	3.28	1.61
Susp Pos Rear	20.23	18.39	4.16	3.29
Susp Pos Front	189.75	116.88	7.04	6.57
Throttle Pos	187.64	154.16	7.55	6.45
Brake Front Pos	0.65	0.45	0.34	0.29

The decision to divide the track into more detailed segments was validated by comparing average performance evaluation metrics. Table 3 illustrates the enhancement of approximately 46% in MSE and 30% in MAE. Furthermore, the computational efficiency improved by close to 50%, further supporting the efficacy of the proposed approach.

7.4 Appendix D

The integration of a polynomial regression model within the random forest structure, discussed in Section 3.4.2, was unsuccessful. To validate the decision to exclude the polynomial model entirely, it underwent a standalone analysis. Figure D-1 illustrates the application of the polynomial regression ML model at predicting correlated speed. In this particular instance, the polynomial was raised to the power of five.

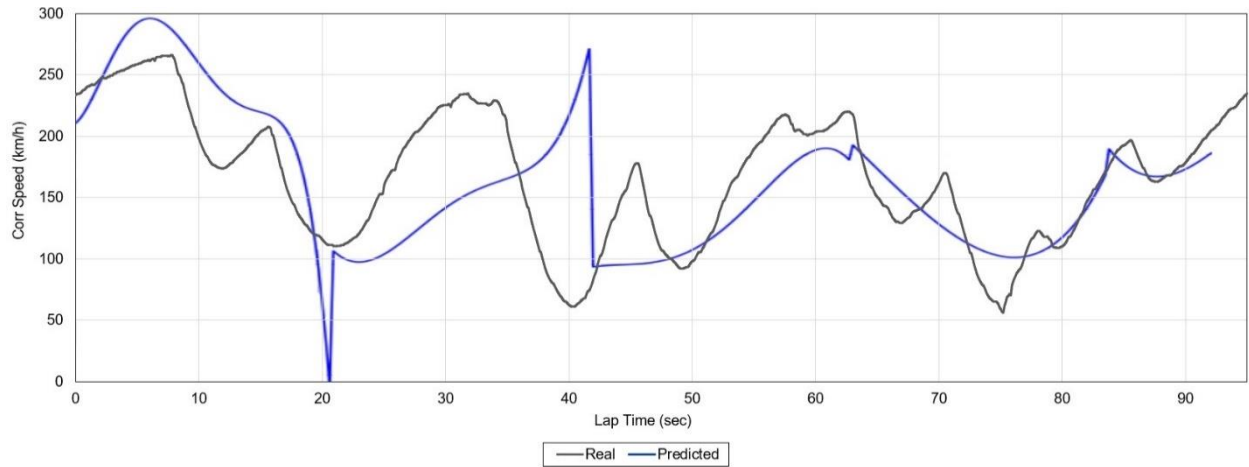


Figure D-1: Correlated speed comparison between real and predicted using a polynomial regression ML model.

The predictions were notably distant from actual observations. Even with coefficients greater than 25, it was impossible to capture the complex dynamics of certain channels, notably those with pronounced noise, such as suspension position and engine speed. Accordingly, this evaluation highlighted both the overwhelming computational demands and the poor performance of the hybrid model.

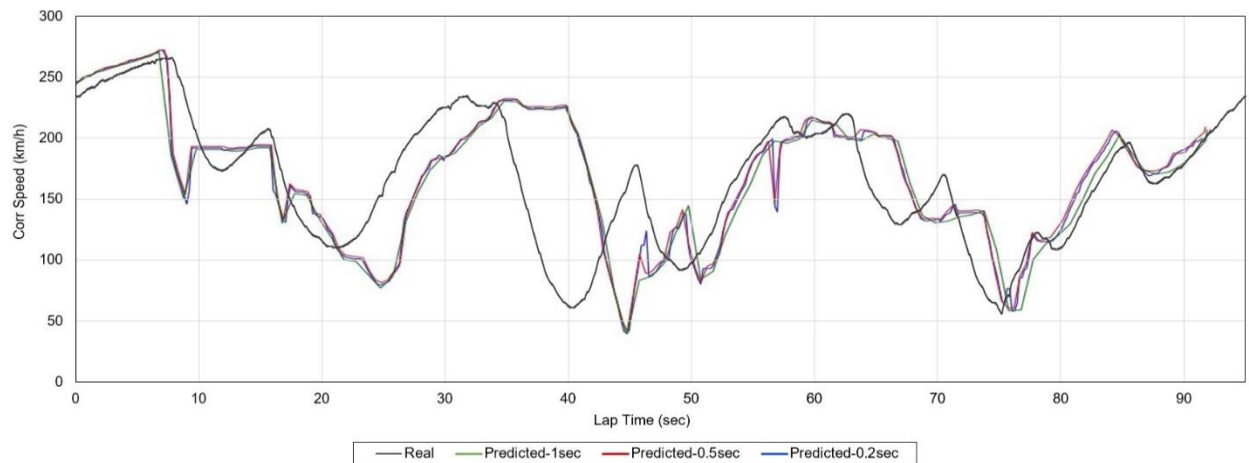


Figure D-2: Time interval comparison of the ML model

In the process of refining the telemetry prediction model, various time intervals were evaluated for their efficacy. As explained in Section 4, when there are shorter time intervals, the ML model must predict a greater number of data points, which can increase the computational demand. Figure 2 shows that a small time stamp, such as 0.2 seconds, also introduces noise into the dataset. Therefore, more training data would be required to ensure that the model is highly

accurate. On the other hand, a time stamp of 1 second proved too broad, omitting crucial features of the predictions. Taking into account the specific constraints of this study, a 0.5-second interval was found to be the best compromise between accuracy and precision.

7.5 Appendix E

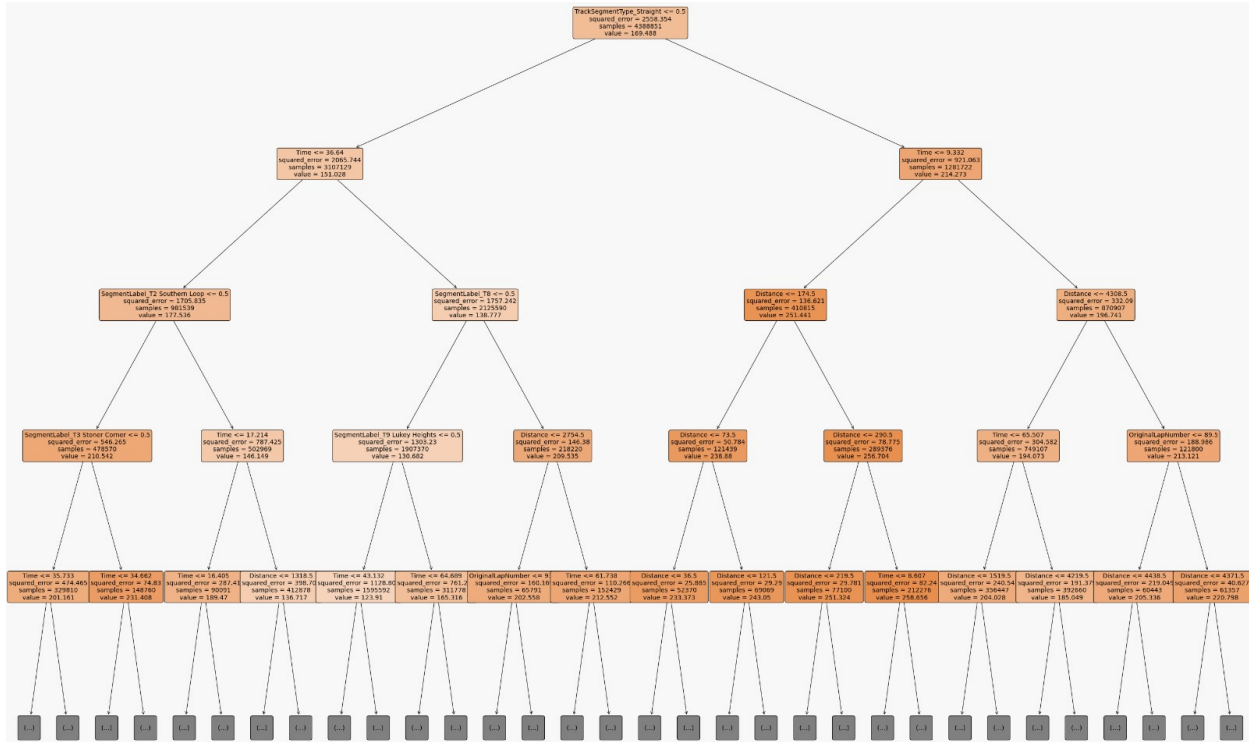


Figure E-1: 1st decision tree of the gear channel from the telemetry prediction ML model.

The study determined that the RF model employed for telemetry channel predictions performed optimally when configured with 15 trees. Each of these trees utilized a bootstrapping strategy, selecting a random subset of the training data for processing. Figure E-1 illustrates the decision-making journey of the first tree of the gear position performance channel within this telemetry prediction ML model. The "max_depth" parameter was configured without constraints, "None". As a result, trees were able to grow to the deepest levels of granularity. The growth, however, was not unbounded. With the "min_samples_leaf" parameter set to 1, trees were designed to undergo node splitting until each leaf node contained only one sample. In addition, with "min_samples_split" set to 2, any split that would result in fewer than two samples was avoided. With each tree producing a unique decision path, the aggregate decisions of the 15 trees resulted in a comprehensive and resilient prediction.

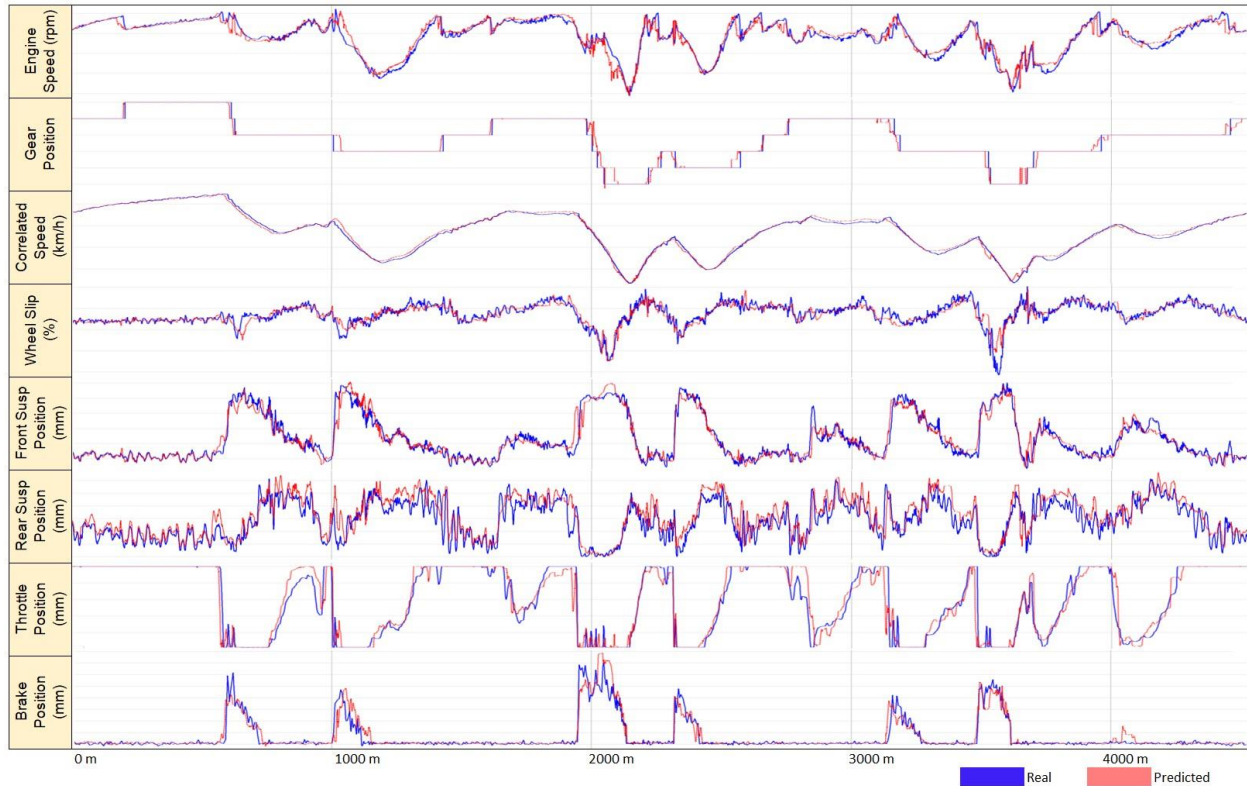


Figure E-2: Additional results of the telemetry prediction ML model without filters.

A comprehensive performance comparison is provided in figure 1 between the predicted lap record (red) and the fastest actual lap (blue) from the analyzed session. In this particular case, the ML model aimed to improve the lap time by 0.405 seconds. Although there was some noise in the predictions, the model's intentions were clear. For example, it suggested smoother throttle control to maintain a higher average speed through corners. Notably, the ML model also favored throttle blipping during downshifts, an action designed to elevate the engine speed and prevent rear wheel lock-up. Such a nuanced prediction further emphasized the ML model's attention to detail. To achieve equilibrium between small features, such as throttle blipping, and unrealistic behaviors, such as gear selection, weighted filtering was applied to specific channels. Among these were engine speed, gear position and correlated speed.

An addition to the telemetry prediction ML model was the extraction of the ideal lap from the analyzed session and overlaying it with the fast lap. Similar to the lap time prediction ML model, a validation process was used to ensure the accuracy of sector extraction by the algorithm.

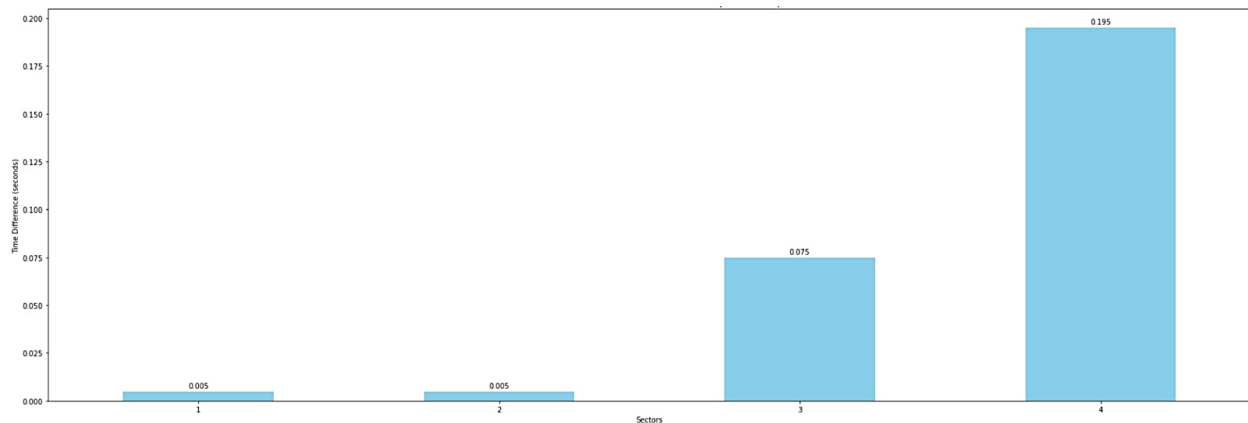


Figure E-3: Difference between ideal and fast lap time computed by the telemetry prediction model.

Table E-3: MoTeC split times report for idea lap time telemetry prediction model validation.

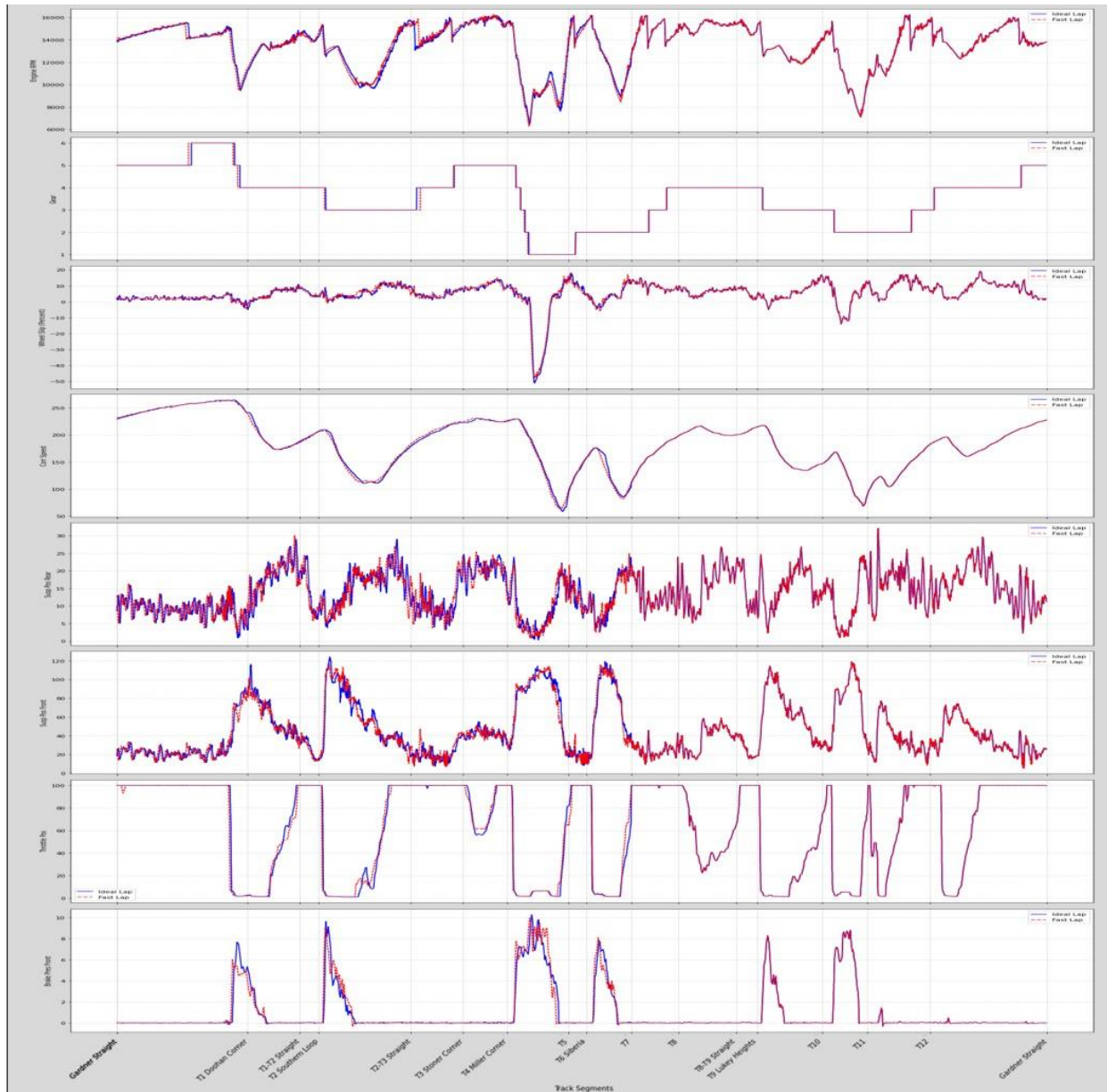
Phillip Island, Test 3 , ZX6R 2/18/2013, Outing 1 9:29:05 AM,															Edictic	Rolling Minimum
Lap 1	Lap 2	Lap 3	Lap 4	Lap 5	Lap 6	Lap 7	Lap 8	Lap 9	Lap 10	Lap 11	Lap 12	Lap 13	Lap 14			
1	28.668866	28.054983	27.115196	26.860293	26.725438	26.727003	26.668779	26.743954	26.869788	26.665101	26.668225	26.718380	26.697593	27.067245	26.665101	26.668779
2	28.156869	28.358501	26.232288	25.773834	25.525076	25.591145	25.334702	25.730267	25.558058	25.433896	25.680449	25.549841	25.584748	25.485988	25.334702	25.334702
3	20.888155	21.015989	20.004706	19.388211	19.303988	19.481811	19.238625	19.640109	19.254435	19.313640	19.255726	19.342535	19.169673	19.419442	19.169673	19.238625
4	27.605109	26.734527	26.607810	26.395663	26.069498	26.164040	26.147894	26.359670	26.327719	26.299362	26.186600	26.239244	26.198985	26.216325	26.069498	26.147894
Totals	105.319000	104.164000	99.960000	98.418000	97.624000	97.964000	97.390000	98.474000	98.010000	97.712000	97.791000	97.850000	97.651000	98.189000	97.238974	97.390000

A session from the testing set was randomly chosen and its split track report is presented in Table E-1. Compared with the results shown in Figure E-3, the telemetry prediction model accurately identified the optimal lap for each sector and the time deviations were within a narrow range of 0.05 seconds.

7.6 Appendix F

Hello Rider, I'm your digital telemetry assistant. I am here to help you improve your performance by analyzing your telemetry data and providing actionable feedback. Now, let's review your telemetry data. From the data, your 'fast lap' time is 94.459 seconds, while your 'ideal lap' time is 94.260 seconds. This means there is a potential for improvement of 0.199 seconds. Breaking it down sector-wise, the time differences are as follows: - Sector 1: You are slower by 0.054 seconds - Sector 2: You are slower by 0.135 seconds - Sector 3: You are faster by 0.005 seconds - Sector 4: You are faster by 0.005 seconds Looking at specific segments, the notable differences are: - 'T4 Miller Corner': You are slower by 0.065 seconds - 'T3 Stoner Corner': You are faster by 0.015 seconds - 'T5': You are faster by 0.035 seconds - 'T2-T3 Straight': You are faster by 0.07 seconds Based on this data, here are some suggestions: 1. Work on your speed in Sector 1 and 2, as you are slower in these sectors compared to your ideal lap. 2. Maintain your speed in Sector 3 and 4, as you are faster in these sectors. 3. In 'T4 Miller Corner', try to improve your cornering speed. You might need to adjust your braking point or line through the corner. 4. Keep up the good work in 'T3 Stoner Corner', 'T5', and 'T2-T3 Straight'. You are faster in these segments. Remember, these insights are based on the data we have. We will continue to monitor your performance and provide more detailed feedback in the future. Keep pushing and let's aim for that ideal lap time!
--

Figure F-1: 1st part of ideal against fast lap AI analysis from QP session.



Brake Analysis: The brake pressure front data shows that the rider applied the brakes more aggressively during the fast lap, with a higher peak pressure. This could suggest that the rider was braking later and harder into the corner during the fast lap, which could have contributed to the higher minimum.

Throttle Analysis: Starting with the throttle application, the rider applied the throttle for a slightly longer duration during the fast lap (8.26 seconds) compared to the ideal lap (8.20 seconds). This suggests that the rider was more aggressive in accelerating during the fast lap. However, the throttle distance remained the same for both laps at 289.0 meters, indicating that the rider maintained a consistent line through the corner in both laps.

Gear and Engine Speed (RPM) In terms of speed, the average speed difference between the ideal and fast laps was minimal at 1.00 km/h. However, the top speed achieved during the fast lap (229.80 km/h) was slightly higher than the ideal lap (229.40 km/h), suggesting that the rider was able to carry more speed into the corner during the fast lap. Conversely, the minimum speed during the fast lap (63.30 km/h) was higher than the ideal lap (58.70 km/h), indicating that the rider was able to maintain a higher speed through the apex of the corner during the fast lap.

Suspension Analysis: Analyzing the suspension data for the 'T4 Miller Corner' segment, we can see that the maximum travel for the front suspension is slightly higher in the fast lap compared to the ideal lap. This could indicate that the bike is experiencing more weight transfer to the front during braking or cornering. On the other hand, the maximum travel for the rear suspension is lower in the fast lap compared to the ideal lap. The differences in suspension travel between the fast and ideal laps could also be related to the speed. The 'Corr Speed' data shows that the rider is carrying more speed into and through the corner in the ideal lap, which could explain why the suspension is working more (i.e., traveling more) in that lap.

Figure F-2: 2nd part of ideal against fast lap AI analysis from QP session.

Let's start with the overall comparison. Your ideal lap time is 94.260 seconds, while the lap record stands at 93.072 seconds. This means there's a potential for improvement of 1.188 seconds. Now, let's break it down sector by sector:

- Sector 1: Your time is 0.332 seconds slower than the lap record.
- Sector 2: You're lagging by 0.308 seconds.
- Sector 3: The gap is 0.230 seconds.
- Sector 4: You're behind by 0.318 seconds.

This data shows that your performance in Sector 4 needs the most improvement.

Let's further break down Sector 1 into its segments:

- 'Gardner Straight': You're 0.112 seconds slower.
- T1 Doohan Corner: You're behind by 0.060 seconds.
- T1-T2 Straight: The gap is 0.021 seconds.

From this, we can see that the 'Gardner Straight' is where you lose the most time in Sector 1. In our subsequent conversations, we'll delve deeper into specific metrics and strategies to improve your performance in these areas. Remember, our goal is to beat the lap record and every fraction of a second counts.

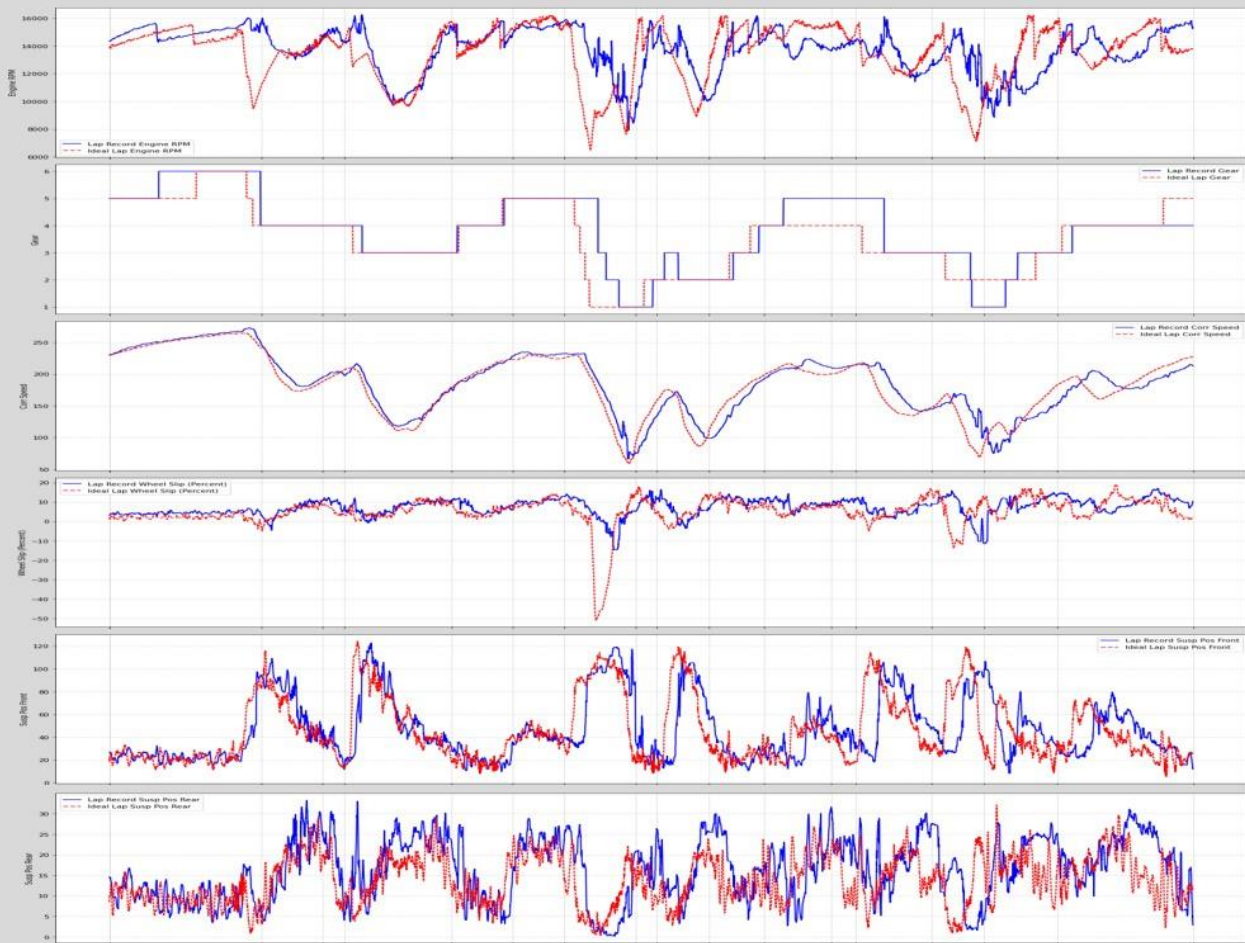


Figure F-3: 1st part of ideal against predicted lap AI analysis from QP session.

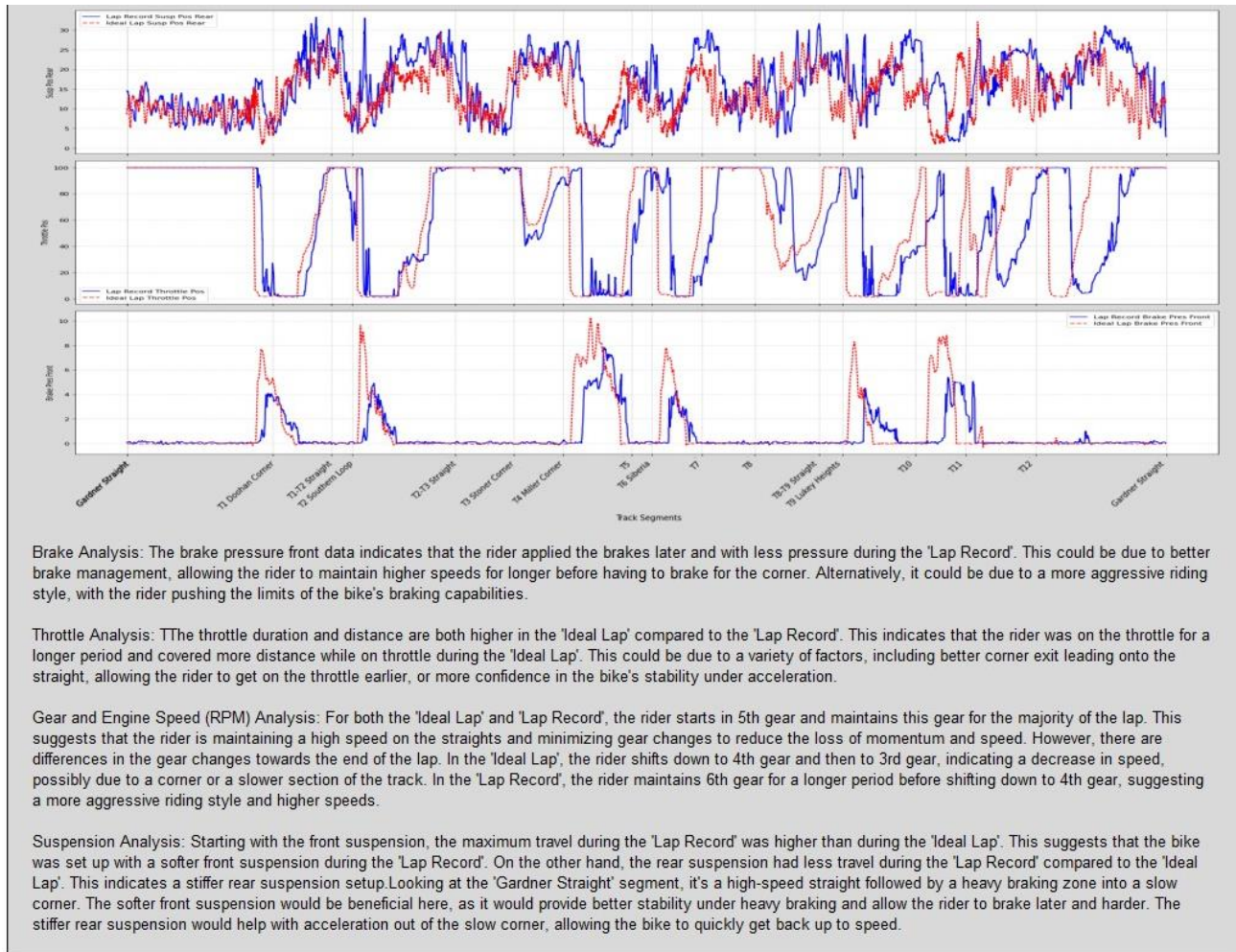


Figure F-4: 2nd part of ideal against predicted lap AI analysis from QP session.

A second validation of the Ai assistant was obtained following FP3. In this qualifying practice session (QP), the rider actually achieved a slower lap time, as shown in Figures F-1 and F-2. The model has, however, learned from its previous sessions and has proposed a different approach to achieving the lap record, as shown in Figures F-3 and F-4.