Utilizing Telemetry Data for Machine Learning-Driven Lap Time Predictions and Race Strategies

Arjun Krishnan, Benjamin Noe, Ajay Patel

May 13, 2024

Abstract

In Formula 1, the difference between winning and losing often comes down to strategic decisions made before the cars even hit the track. This project develops a machine learning model using the FastF1 Python library to predict lap times and optimize tyre strategies, paving the way for smarter race planning. Our model utilizes rich telemetry and timing data to forecast lap times accurately, allowing us to simulate different tire strategies and identify the most beneficial ones under varying conditions.

This work is the first step towards a broader goal: a comprehensive pit stop strategy model. We aim to make advanced race strategy accessible, providing a tool that is easy to use for anyone. By breaking down complex data and offering clear, actionable insights, we are setting the stage for more informed, data-driven decisions in Formula 1.

1 Introduction

Inspired by the innovative "Virtual Strategy Engineer" paper[HTGB20], our project introduces a user-friendly race simulation model to a wider audience. Heilmeier et al.'s research in utilizing machine learning and neural networks to automate race strategies inspired us to simplify this technology. Their approach showed that leveraging artificial neural networks could predict and optimize race outcomes effectively, considering elements like tyre wear and fuel levels. The paper employs a confusion matrix to determine how accurate the model is by looking at whether the model was able to correctly predict when a car pitted in a race. However, correctly predicting when a car pitted in a race does not necessarily translate to the truly best and correct time a car should have pit. Our goal is to solve for this issue, creating a simplified, yet accurate model that is able to find the true correct pit stop time for any given car.

To accomplish this lofty goal, we outlined several steps we needed to take to build such a model. First,

we wanted to create a lap time simulator trained on historical data so that we could directly compare different tyre strategies. Next, we wanted to use the lap time simulator to create a more complex race simulator engine that takes into account all 20 drivers. Finally, we aim to use the race simulator engine along with historical data to see how race results would change when different strategies are employed. This would give us a better sense of how "correct" our model is. With this approach in mind, this paper focuses specifically on the first step in this project, examining our lap time simulator and its applications.

2 Data Preparation

To create a lap time simulator, we again took inspiration from "Virtual Strategy Engineer." However, rather than creating our own dataset or using their own, we opted to pull all our data from the FastF1 Python Package. FastF1 contains a vast amount of F1 data from 2018-2024 from event level data to granular telemetry data. In examining the timing data, with the project requirements in mind, these were the final attributes we settled on for our model.

Feature Name	Definition
Year	Year the race occurred
RaceName	Track where the race occurred
Driver	The name of the driver
Team	The name of the team
Lap Number	What number lap it is in the race
InLap	If the car is pitting on a given lap
OutLap	If the car is exiting the pits on a given lap
Compound	The type of tyre being used
TyreAge	The age (in lap numbers) of the tyre being used
Position	What place the racer is in
TrackStatus	State of the track (Clear, VSC, Red Flag, etc.)
LapTime	The time (s) it took to complete a lap

Table 1: Independent and Dependent Variables.

To acquire this data, we created a data acquisition loop that iterated through each event in each season from 2018-2023. From there, we pulled every single lap in a given race containing the data from our feature set above and appended it to a compiled dataframe. From these laps, some data cleaning steps were applied. All wet races were removed from the dataset due to our belief that they would not accurately be captured by the model, and similar being done in other studies. Next, any laps that were missing a lap time were deleted, as they would prove of no use in a model made to predict lap time. At the end of this process, we were left with a dataset containing 124,756 total laps.

3 Modeling

To develop a robust predictive model for our lap time simulation, we utilized ensemble learning methods, which combine multiple individual models to enhance predictive performance and stability. Specifically, we implemented a Voting Regressor, integrating two powerful regression algorithms: Extra Trees Regressor and Random Forest Regressor.

3.1 Model Setup

The Voting Regressor operates by averaging the predictions of its constituent regressors. For our implementation, we chose:

- **Extra Trees Regressor:** An ensemble method like Random Forest but creates trees from the entire dataset instead of a bootstrap sample and uses random thresholds for splitting nodes.
- **Random Forest Regressor:** A well-known ensemble method that builds multiple decision trees and merges them together to obtain a more accurate and stable prediction.

Each regressor in our ensemble was initialized with default parameters, except where randomness is controlled for reproducibility (e.g., setting random state in Random Forest Regressor).

3.2 Data Preparation and Training

Our dataset comprises various features, including year, race track, driver details, lap information, tire usage, and track conditions. The data was split into a training set and a test set with an 80/20 ratio using a stratified method based on the year and race to ensure a representative distribution of data across both sets.

The model was trained on the training data using 5-fold cross-validation to ensure the model's generalizability and to mitigate overfitting. The median absolute error was used as the scoring function during cross-validation, as it provides a clear measure of prediction accuracy without being as affected by outliers compared to other common accuracy metrics.

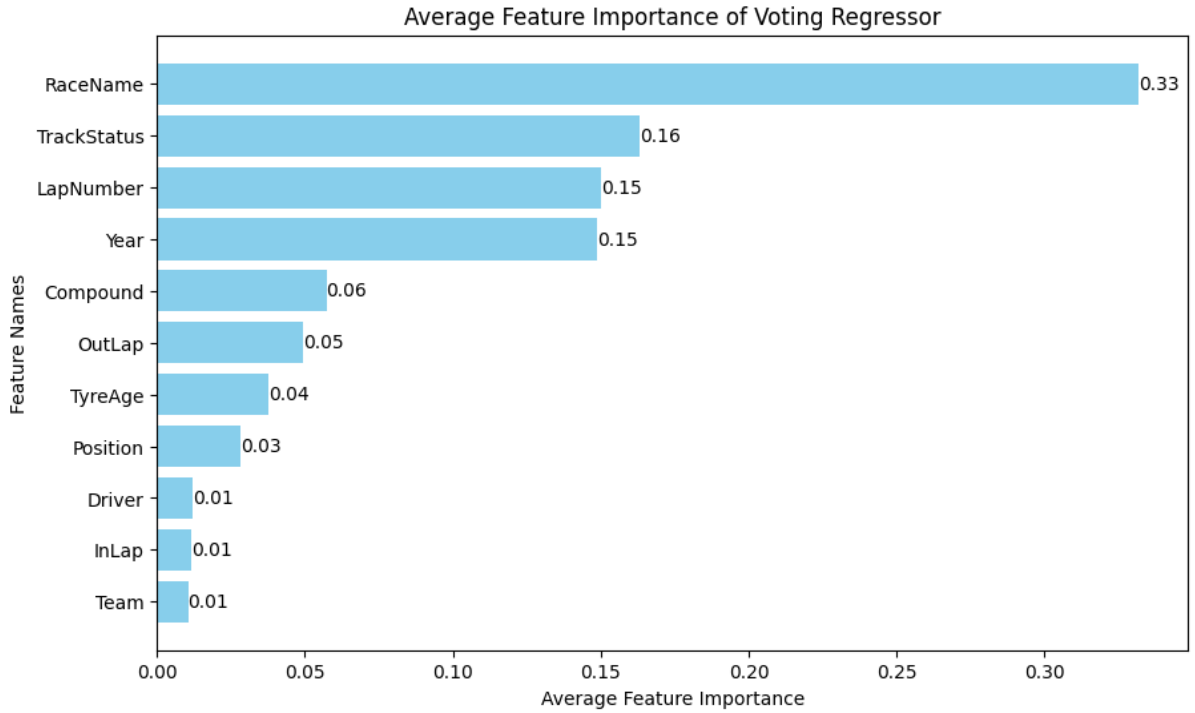

Figure 1: Model Feature Importance.

3.3 Model Evaluation

After training, the Voting Regressor was used to predict lap times on the unseen test set. The model's performance was evaluated using two metrics:

- **Median Squared Error:** Measures the median of the squares of the errors, i.e., the median squared difference between the estimated values and the actual value.
- **Median Absolute Error:** Provides a measure of errors between paired observations. It's particularly useful because it provides a linear score without squaring each error, giving a more direct understanding of error magnitude.

The results from the test set indicated that the ensemble model performed well:

- **Test Median Squared Error:** 0.197
- **Test Median Absolute Error:** 0.443
- **R-Squared:** 0.965

The R-squared value of 0.965 shows that the model explains 96.5 percent of the variance in the lap times, highlighting its strong predictive capabilities.

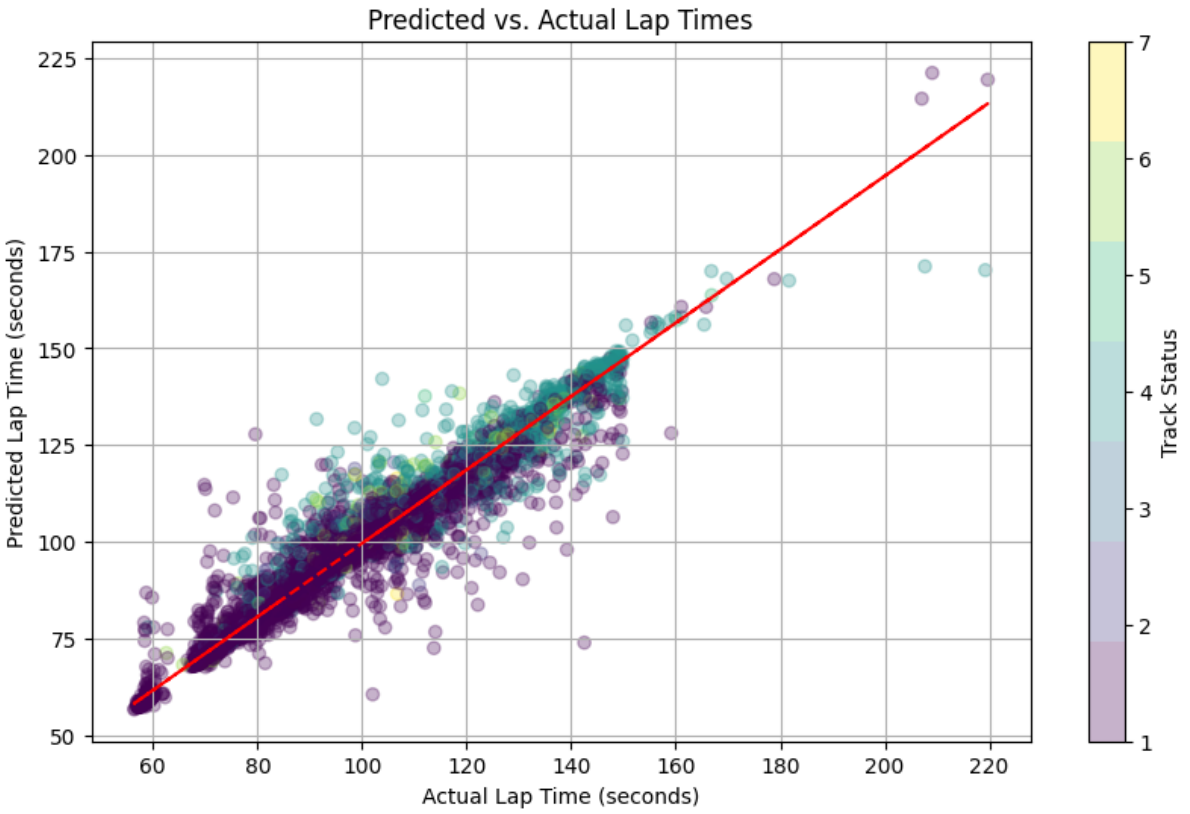

Figure 2: Predicted vs. Actual Lap Times.

3.4 Cross-Validation Results

The cross-validation results further validated the model's performance across different subsets of the training data, showing its consistency and stability under varying conditions. Here are the detailed cross-validation Mean Absolute Error scores:

- Cross-Validation Median Absolute Error Scores: [0.500325 0.49068 0.482155 0.49155 0.4797125]
- Mean Cross-Validation Median Absolute Error: 0.489

These scores demonstrate the model's reliability, with an average Median Absolute Error close to the single-test Median Absolute Error, confirming the effectiveness of our modeling approach. An overall Cross-Validation score of 0.489 means that we can expect a median absolute error of 0.489 seconds. While this value may seem a little large, by examining the Median Absolute and Median Squared Error by lap time, we can see that the first couple of laps, due to a chaotic track environment, and the last few laps, due to smaller sample sizes, have larger Median Absolute and Squared Errors, which logically makes sense and cause the overall Median Absolute Error to be greater.

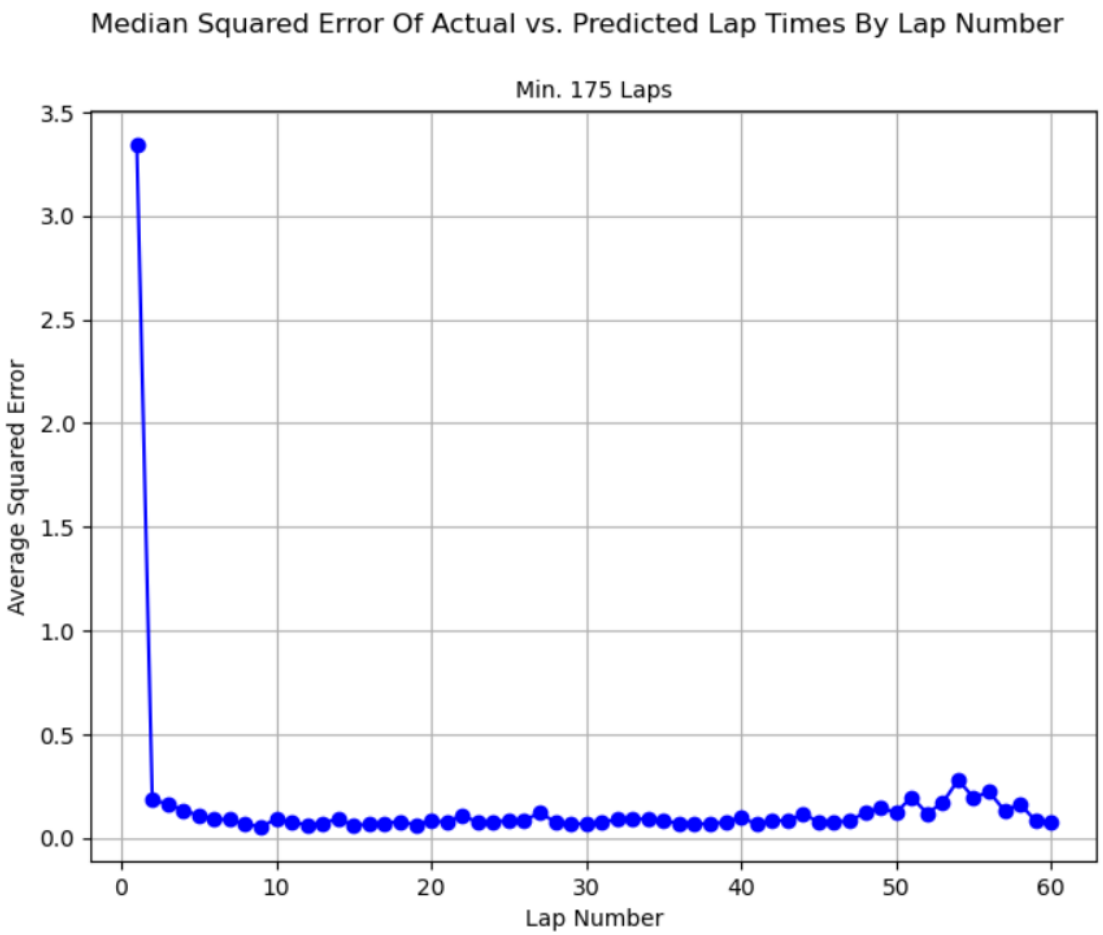

Figure 3: Median Squared Error by lap.

4 Results

With our model complete, we wanted to do some exploratory analysis to examine the results of the model. First, we decided to look at the specific race tracks where our model is more successful.

Table 2: Top 5 and Worst 5 Tracks Based on Deviation Percentage

RaceName	**Avg Actual Lap Time**	**Avg Predicted Lap Time**	**Deviation**	**Percentage Deviation**
Austin	93.3734	93.3677	-0.0057	0.0061%
Melbourne	88.0677	88.0812	0.0135	0.0153%
Imola	99.9351	99.8916	-0.0435	0.0436%
Lusail	92.0199	91.9746	-0.0453	0.0492%
Monte Carlo	84.3151	84.2588	-0.0563	0.0667%
		...		
Sakhir	90.7839	90.5682	-0.2158	0.2377%
Portimão	86.5495	86.7648	0.2153	0.2487%
Silverstone	84.0349	84.2607	0.2258	0.2687%
Singapore	88.1284	88.4613	0.3328	0.3777%
Yas Marina	74.9999	75.3479	0.3480	0.4640%

Unsurprisingly, our model has the least amount of deviation for courses that were apart of the F1

calendar for many years. Austin and Monte Carlo for example have been apart of the calendar every year from 2018-2023. At the bottom of our list, we have circuits that tend to be tougher circuits to drive, and thus have more unpredictability. Singapore, for example, is known for its very challenging tight turns causing many crashes and unpredictability. Still, the worst performing track, Yas Marina, only had a deviation percentage of 0.4640 which is still incredibly low.

In our analysis of the predictive model for lap times at the Shanghai Grand Prix 2018, we focused on comparing the predicted times against the actual race data for drivers Pierre Gasly (GAS) and Sergio Perez (PER). Figure 4 illustrates a detailed comparison over the course of 55 laps with the deviations between the model's predictions and real-world outcomes.

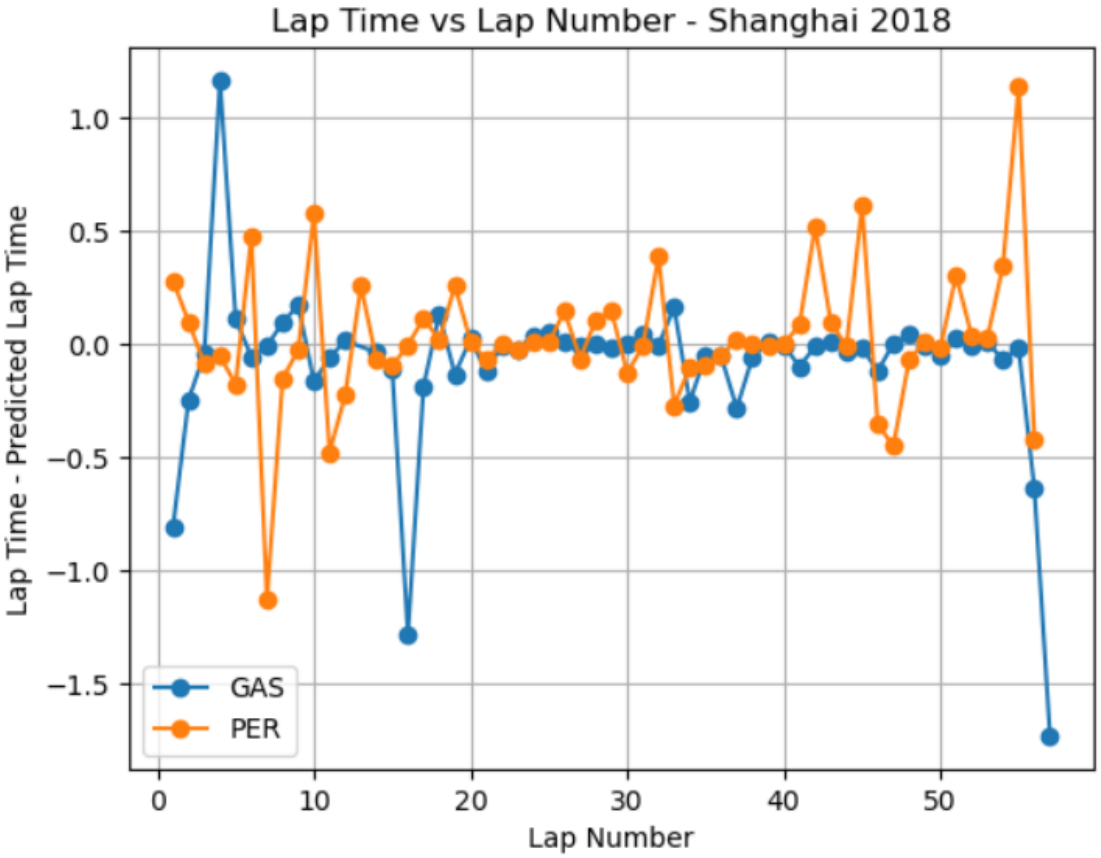

Figure 4: 2018 Shanghai Lap Times Predicted vs. Actual (Gasly and Perez)

The graph shows a stable performance in the predictions for Gasly in the heat of the race, but variations do appear on the first few and last couple laps. However, these variations are nowhere greater than two seconds, which while meaningful, is not way too far different. This indicates a high level of accuracy in the predictive model for Gasly, suggesting that the model is well-tuned to his racing style and the specific dynamics of the Shanghai circuit. It is worth noting that Gasly was penalized ten seconds

at his final race time for causing an avoidable collision, but this is not reflected in individual lap times, therefore it does not show up in our model.

On the other hand, the predictions for Sergio Perez exhibit a more significant variance, particularly noticeable at around the last few laps where the prediction consistently deviates and at the start of the race as well. The model could be less familiar with Perez's racing style than compared to Gasly's, or it could simply be due to random chance that the predictions exhibit more deviation. Again, the deviations are nowhere more than a max of two seconds, showing our model's overall strong ability to predict lap times.

5 Simulation

The enhanced predictive capabilities demonstrated in the "Results" section enable us to simulate race outcomes with a high degree of accuracy using our advanced race simulation model. This model has been designed to incorporate various inputs to generate a detailed simulation of any given race.

5.1 Model Inputs

The primary inputs for the simulation model are:

- **Driver and Team Name:** By specifying the driver, the model adjusts for individual driving styles, historical performance metrics, and specific strengths and weaknesses that influence race outcomes. It also allows the simulation to account for team-specific variables such as pit stop efficiency, team strategies, and the quality of the car setup.
- **Year:** This input is important as it helps let the model know the strength of the given team and car for a particular year.
- **Race Name:** This input is crucial as they provide context regarding the track layout, typical weather conditions, and other race-specific factors that were present during the specified year.

5.2 Practical Applications

This simulation tool is invaluable for teams during the pre-race strategy formulation. It allows teams to explore various "what-if" scenarios, tailoring strategies to maximize performance based on predicted

outcomes. Furthermore, it serves as a training tool for drivers, providing them with virtual experience under a wide range of race conditions. Below, we simulated every possible pit stop strategy for Pierre Gasly for the 2018 Chinese Grand Prix.

Table 3: Shanghai 2018 Optimal Racing Strategies and Pit Laps Analysis

Final Time (s)	**Pit Lap(s)**	**Strategy**
5330.26691	17	[MEDIUM, HARD]
5330.68597	19	[MEDIUM, HARD]
5330.90684	18	[MEDIUM, HARD]
5331.03798	21	[MEDIUM, HARD]
5331.19960	16	[MEDIUM, HARD]
5331.23530	20	[MEDIUM, HARD]
5332.11906	16	[SOFT, HARD]
5332.31504	22	[MEDIUM, HARD]
5332.35340	17	[SOFT, HARD]
5333.23397	18	[SOFT, HARD]
	...	
5518.92802	[10, 11]	[HARD, SOFT, SOFT]
5502.17034	[10, 11]	[MEDIUM, SOFT, SOFT]

As we can see, the model identifies two clear dominant strategies: starting on the medium compound and pitting to the hard compound, or starting on the soft compound and likewise pitting to the hard compound. Using this, we could strategize for Pierre Gasly to start on the medium tyre with a pit window of lap 17-22 and his teammate at the time, Brendon Hartley, to start on the soft tyre with a pit window of 15-18 to account for possible deviation. This simulation tool provides an accurate way to derive tyre strategies for any given race.

In addition, we wanted to run some tests on 2024 data. We simulated the pit stop strategies for Lando Norris.

Table 4: Lando Norris Miami 2024 Optimal Racing Strategies and Pit Laps Analysis

Strategy Rank	**Pit Lap(s)**	**Strategy**
1	28	[MEDIUM, HARD]
2	29	[MEDIUM, HARD]
3	30	[MEDIUM, HARD]
4	31	[MEDIUM, HARD]
5	27	[MEDIUM, HARD]

Interestingly enough, our model predicted the quickest strategy to be starting on mediums, pitting on lap 28, and switching to the hard tyre. This strategy is the exact strategy Lando Norris and McLaren used during the 2024 Miami Grand Prix that led to Lando's first ever F1 win. This example shows the models effectiveness and how it can be used to accurately predict lap times and determine strategies.

6 Conclusion and Future Work

This study has presented a simplified, yet comprehensive approach to predicting Formula 1 lap times and optimizing race strategies using advanced machine learning techniques. By leveraging the FastF1 library, we have accessed rich telemetry and timing data, which served as the foundation for our predictive models. Our analysis has demonstrated that the models can not only forecast lap times with high accuracy but also simulate race strategies that can significantly influence race outcomes.

The application of the Voting Regressor, combining the strengths of Extra Trees and Random Forest regressors, has proven effective in handling the complex and dynamic nature of F1 racing data. The model's ability to account for many variables such as tyre wear and track characteristics has allowed for predictions that are relevant to real-world racing scenarios.

6.1 Limitations and Future Work

While our model demonstrates strong predictive capabilities, it is not without limitations. The accuracy of the predictions can be affected by random events such as accidents or sudden weather changes which are inherently unpredictable and not always reflected in historical data.

For future work, we aim to:

- Incorporate real-time data feeds during races to update predictions dynamically as race conditions change.
- Expand the model to include more granular data on track and tire conditions to enhance the accuracy of tire strategy predictions.
- Expand upon this project and create a full race simulator, taking into account all drivers at once.
- Create a robust pit stop predictor based on the full race simulator.

In conclusion, combining machine learning with Formula 1 racing can create opportunities to optimize race strategies. As we keep enhancing and expanding upon our models, the possibilities for transforming racing strategies are becoming more and more real. This research boosts the strategic play in one of the most high-tech sports on the planet where each tenth of a second is meaningful.

References

[HTGB20] Alexander Heilmeier, André Thomaser, Michael Graf, and Johannes Betz. Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport. *Applied Sciences*, 10(21), 2020.