Department of Informatics University of Leicester

CO7201 Individual Project

Interim Report

AI/ML-Based Formula 1 Race Outcome Prediction Using Historical and Real-Time Data

Tarun Datta

td188@student.le.ac.uk

249033956

Project Supervisor: Dr Furqan Aziz

Second Marker: Dr Karim Mualla

Word Count: 836

25th July 2025

Declaration

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Tarun Datta

25/7/2025

Contents

1. Introduction

This report outlines the current progress on the MSc dissertation titled **AI/ML-Based Formula 1 Race Outcome Prediction System**. The project aims to predict driver finishing positions using machine learning, leveraging both historical and real-time data. It addresses feedback from the preliminary report by clarifying technical components, refining the project scope, and updating the project timeline.

2. Status of Requirements

Each paragraph in this section states the previously mentioned essential requirements and then details their status. The requirements previously listed as recommended and optional have not been started yet, and thus not discussed in this report.

3. Completed Tasks

3.1 Data Collection & Preprocessing

Data from the 2022–2025 Formula 1 seasons was collected using the FastF1 API. A custom pipeline was built to clean and prepare the data, including outlier removal, converting team/driver names into usable formats, and combining lap-level data into broader indicators like momentum and consistency. Weather factors (e.g., rain chance, DNF risk) were also integrated. The processed data is stored in an SQLite database via Django ORM, with caching to minimize repeated requests. Warnings and issues during preprocessing are auto logged for easier debugging.

3.2 Web Platform

A complete web platform was developed using Django, styled with TailwindCSS and JavaScript. It includes secure login, registration, and email verification. Logged-out users see a blurred prediction page prompting sign-in. The prediction page supports model selection via dropdown, while other pages offer race result search (2022–2025) and live standings for drivers and constructors. The site is responsive, with mobile compatibility and dark mode for improved user experience.

3.3 Model Development

Model	MAE	R²	Notes
Ridge	1.79	0.77	Captures consistency, reliability
XGBoost + Ridge	1.18	0.91	Uses Bayesian-tuned XGBoost stacked with Ridge

Two machine learning models were developed to predict driver finishing positions, evaluated using Mean Absolute Error (MAE) — the average difference between predicted and actual positions (lower is better) — and R², which measures how well the model explains variation in race outcomes (higher is better; 1.0 indicates perfect fit).

The first model, Ridge Regression, is a linear approach that captured consistent patterns like team reliability and driver stability. It achieved an MAE of 1.79 and an R^2 of 0.77, offering a strong baseline and demonstrating the value of regularized linear models in racing contexts.

The second model was an ensemble combining Ridge Regression with XGBoost, a nonlinear algorithm capable of learning more complex patterns such as race-specific dynamics or unpredictable weather effects. Using stacking, this ensemble leveraged the strengths of both models. XGBoost was fine-tuned via Bayesian optimization to improve accuracy.

This combined approach reduced the MAE to 1.18 and raised R^2 to 0.91, indicating strong predictive ability across unseen races. Key features included rivalry performance (6.13), circuit affinity (1.00), and teammate battle (0.81), emphasizing the role of both individual performance and inter-driver dynamics.

4. **In Progress**

4.1 Circuit-Type-Based Optimization

Racetracks differ widely in characteristics—some emphasize high speed, others feature tight corners or frequent overtaking zones. To account for this, the project is testing a circuit grouping strategy based on shared traits like layout and overtaking difficulty. Techniques such as KMeans clustering and PCA are being explored to group circuits more effectively.

Model performance is being tested across distinct circuit types—like Monza (high-speed), Monaco (tight street), and Silverstone (balanced)—to evaluate prediction accuracy within each group. The aim is to adjust model behavior based on circuit category and reduce per-track prediction error.

4.2 LightGBM Implementation and Tuning

LightGBM, a fast and efficient tree-based algorithm, is being integrated to enhance prediction accuracy. Building on insights from Ridge Regression and XGBoost, it is well-suited for handling large, complex datasets.

The model is currently undergoing hyperparameter tuning, with added features such as driver experience and performance trends under evaluation. If results are strong, LightGBM may complement or even replace parts of the current ensemble setup.

5. Pending Tasks

5.1 Dashboard Visualizations

To improve user engagement and model clarity, the platform will include interactive charts. Lap time distributions—visualized using Matplotlib—will help users spot performance trends and outliers. Additionally, driver comparison tools built with Plotly will allow users to view head-to-head statistics, especially between teammates or key rivals, across multiple races.

5.2 Live Data Integration

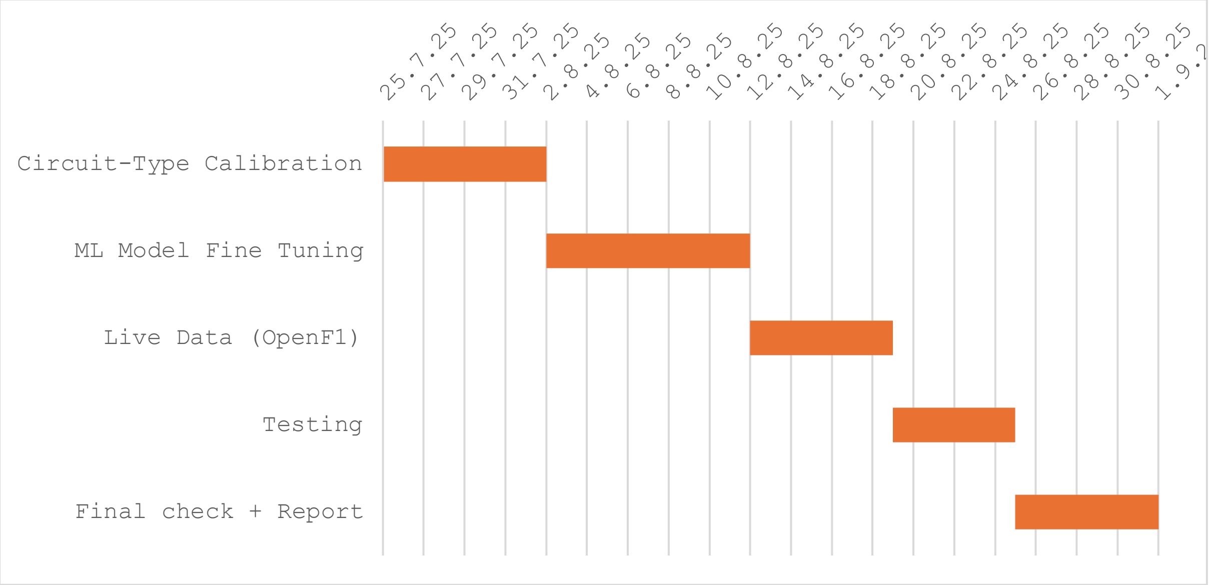
F1 races often change unexpectedly due to real-time factors like tire wear, sudden weather shifts, or driver retirements (DNFs). These can drastically impact race outcomes—for instance, unexpected rain may disrupt strategy or force extra pit stops.

To address this, a live data system is being integrated using OpenF1, supported by a FastAPI backend. Real-time updates will be delivered via WebSockets for fast-changing data (like position and tire conditions) and REST API for periodic summaries (e.g., lap times, session updates). This setup allows the model to adjust predictions as the race unfolds, providing more accurate and context-aware outputs during live events.

5.3 Final Polish

With key features in place, remaining efforts will focus on improving usability and design. Planned updates include a cleaner results page, smoother leaderboard transitions, and general UI enhancements to ensure a more seamless and responsive experience for users.

6. Revised Timeline



7. Reference

- [1] FastF1 Python API – Historical data extraction.

- [2] OpenF1 API – Real-time driver & weather telemetry.
- [3] Groll et al., 2019 – Sports regression modeling.
- [4] Bell, T. (2021) – Feature engineering in motorsports.
- [5] Dubois & Patel (2022) – Track-specialized models.
- [6] Nielsen (2018) – Rain/weather implications in race outcomes.
- [7] Rossi (2020) – Confidence evaluation in high-variance sports.