A Term Paper Report

On

Smart Telemetry Analysis & Lap Time Predictor for F1

Submitted to

Amity University Uttar Pradesh

in partial fulfilment of the requirements for the award of the degree of
Bachelor of Technology in
Computer Science & Engineering

Submitted By

Student Name: Abhinav Shukla
Enrollment. No.: A7605223121

under the guidance of
Dr. Sachin Kumar
Assistant Professor
Department of Computer Science and Engineering

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH
LUCKNOW (U.P.)
July 2025

AMITY UNIVERSITY

UTTAR PRADESH

DECLERATION BY THE STUDENT

I, Abhinav Shukla, student of B. Tech. hereby declare that the project titled "**Smart Telemetry Analysis & Lap Time Predictor for F1"** which is submitted by me to Department of Computer Science and Engineering, **Amity School of Engineering and Technology**, Amity University Uttar Pradesh, Lucknow, in partial fulfilment of requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The Author attests that permission has been obtained for the use of any copyrighted material appearing in the Project report other than brief excerpts requiring only proper acknowledgement in scholarly writing and all such use is acknowledged.

Lucknow

Date: 16/07/2025

Abhinav Shukla
B. Tech. (CSE)
A7605223121

AMITY UNIVERSITY

—————UTTAR PRADESH—————

CERTIFICATE

On the basis of the declaration submitted by **Abhinav Shukla,** student of B. Tech., I hereby certify that the project titled **“Smart Telemetry Analysis & Lap Time Predictor for F1”** which is submitted to **Amity School of Engineering and Technology**, Amity University Uttar Pradesh, Lucknow, in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **Computer Science and Engineering** , is an original contribution with existing knowledge and faithful record of work carried out by her under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Lucknow
Date: 16/07/2025

Dr. Sachin Kumar
Supervisor
Assistant Professor
Amity University, Lucknow.

AMITY UNIVERSITY

UTTAR PRADESH

ACKNOWLEDGEMENT

It is a high privilege for me to express my deep sense of gratitude to those entire faculty members who helped me in the completion of the project, especially my internal guide **Dr. Sachin Kumar** who was always there at hour of need.

My special thanks to all other faculty members, batchmates & seniors of **ASET**, Amity University Uttar Pradesh for helping me in the completion of project work and its report submission. I would also like to thank my parents, my brother and my friends for helping me out and keeping me motivated throughout.

Abhinav Shukla

B. Tech. (CSE)

A7605223121

AMITY UNIVERSITY

——————UTTAR PRADESH——————

ABSTRACT

In the high-stakes world of Formula 1 racing, data-driven decision-making plays a critical role in achieving peak performance. Every millisecond on the track is influenced by telemetry data that captures key parameters such as throttle input, gear shifts, speed, and braking behaviour. This project focuses on designing a smart telemetry analysis system that leverages such data to predict lap times, enabling insight into driver performance and car behaviour. By using open-source tools like the FastF1 Python library, real telemetry from official F1 sessions was accessed, cleaned, and transformed into a model-ready dataset.

A machine learning pipeline was developed using the Random Forest Regressor algorithm to predict lap times based on selected telemetry signals. The model was trained and evaluated using standard regression techniques, including the Mean Absolute Error metric. The system also includes visual analytics to interpret driver inputs and model behaviour, allowing for a more intuitive understanding of performance patterns. With appropriate preprocessing and feature selection, the model demonstrated reliable predictions, validating the feasibility of using real-time race telemetry for performance forecasting.

This project bridges the gap between motorsport analytics and practical machine learning by creating an end-to-end system that is both functional and educational. It highlights the growing role of AI and data science in sports, particularly in motorsports where telemetry data is abundant yet complex. The modular design of the system allows for future enhancements such as multi-lap strategy modelling, contextual input analysis (weather, tire degradation), and adaptation for simulation racing environments. Overall, this work showcases how telemetry and predictive analytics can contribute to smarter, data-backed racing insights.

CONTENTS

INTRODUCTION

1.1 Background:

Formula 1 (F1) racing is not only a test of driver skill but also a battleground of advanced engineering and data-driven decision-making. Over the last few decades, the role of telemetry in motorsports has grown exponentially. Every millisecond on track is influenced by massive volumes of data gathered from hundreds of sensors embedded in the car. These sensors provide real-time information regarding speed, throttle input, gear shifts, engine performance, braking patterns, tire wear, and environmental conditions. Teams monitor this data to evaluate car performance, plan race strategies, and gain competitive advantages.

In recent years, access to real F1 telemetry has expanded beyond factory teams. Open-source libraries and tools now empower students, developers, and researchers to work with official F1 data—unlocking opportunities to explore performance trends and build intelligent systems around them. Among these, FastF1**[1]** has emerged as a Python library that provides structured access to official Formula 1 telemetry and timing data.

The increasing availability of motorsport telemetry data**[9]** opens up avenues for practical machine learning applications. Predicting lap times is one such use case that combines data analysis with predictive modelling. Lap-time prediction, when accurate, can be used for driver benchmarking, performance forecasting, and strategy development. While this is a common practice in professional teams, the challenge lies in building a simplified yet effective prediction system using accessible resources.

This project was motivated by the desire to create a tool that analyses telemetry data from F1 races and uses it to predict lap times. The goal is not only to replicate professional workflows but also to understand and demonstrate how raw telemetry can be turned into actionable insights. It brings together software development, data science, and domain knowledge into a unified academic exercise.

1.2 Objectives

The primary objectives of this project are:

1. **Data Acquisition & Preprocessing**: Fetch high-quality telemetry data using the FastF1 library and perform cleaning, filtering, and feature extraction.
2. **Exploratory Data Analysis**: Analyse relationships between key performance indicators—such as throttle usage, speed, braking behaviour, and gear shifts—and lap times.

3. Model Development: Design and train a machine learning regression model (Random Forest **[2, 3]** Regressor) to predict lap times based on selected features.
4. Model Evaluation: Evaluate predictive performance using metrics like Mean Absolute Error **[3]** (MAE) and interpret feature importances.
5. **Visualization & Reporting**: Generate clear plots and graphs to illustrate data patterns, model behaviour, and prediction accuracy.

1.3 Scope of the Project

This project focuses on telemetry data from specific Formula 1 sessions—such as practice or qualifying laps—available via FastF1. The scope includes data cleaning, feature selection, model training, evaluation, and result visualization. External factors like weather changes, tire degradation, or on-track incidents are not explicitly modelled here, though the architecture is designed to allow future extensions.

1.4 Data Science & Machine Learning in Sports Analytics

The field of sports analytics leverages statistical techniques and machine learning **[2, 3]** to extract insights from raw performance data. A typical data-science pipeline comprises:

1. **Data Collection**: Retrieval of raw data—here, telemetry signals such as speed and throttle—using APIs or dedicated libraries.
2. **Data Preprocessing**: Cleaning missing/erroneous values, aligning time series, and engineering new features (e.g., average brake pressure per sector).
3. **Feature Engineering**: Selecting or creating variables that best capture the relationships you want to model (for example, combining gear shifts and throttle usage into a "power usage" metric).
4. **Model Selection & Training**: Choosing an algorithm (e.g., Random Forest) and fitting it to the data to learn the mapping from input features to target outputs (lap time).
5. **Model Evaluation**: Assessing performance on unseen data using metrics such as MAE, Root Mean Squared Error (RMSE), or R^2, and checking for overfitting.
6. **Interpretation & Visualization**: Explaining model behaviour via feature-importance rankings, residual plots, and communicating findings through clear figures.

Applying this pipeline in an F1 context illustrates how machine learning can support tactical and strategic decisions—even in highly dynamic environments.

1.5 Significance of the Study

This study demonstrates how complex, high-frequency sports data can be harnessed in an educational setting to build a fully functional predictive system. It bridges the gap between academic learning in data science and real-world applications in sports engineering, opening avenues for further research in motorsport analytics and dynamic performance modelling.

LITERATURE SURVEY

2.1 Introduction to Telemetry in Motorsport

Telemetry systems in motorsports have evolved from simple lap-time counters to real-time, high-frequency data streams. In Formula 1, telemetry plays a crucial role in monitoring car health, optimizing performance, and crafting strategic decisions. Modern F1 cars are equipped with hundreds of sensors that capture data related to engine output, braking behaviour, throttle usage, gear transitions, steering angle, tire temperature, fuel consumption, and more.

Traditionally, this data has been reserved for factory engineers and race strategists. However, recent advancements and public access initiatives have enabled broader communities—including students and researchers—to access this data through platforms like **FastF1**, offering structured APIs for historical session data, lap timing, and telemetry channels.

2.2 Role of Data Analytics in Formula 1

Data analytics in Formula 1 focuses on uncovering performance insights, minimizing uncertainty, and simulating possible outcomes. McLaren, Mercedes, Red Bull, and other top teams employ advanced simulation platforms powered by statistical models, optimization algorithms, and machine learning pipelines. These systems support race-weekend decision-making such as tire selection, fuel strategy, and overtake planning.

As noted by King and Cornet**[11]** (2021), predictive analytics in motorsport is shifting from simple regression models to hybrid models incorporating AI, neural networks, and real-time feedback mechanisms. While full access to team-specific models remains proprietary, various public studies have showcased how open-source tools can replicate simpler predictive workflows with reasonable accuracy.

2.3 Existing Work on Lap Time Prediction

Lap time prediction has been explored in both academic and enthusiast circles. Early approaches used classical regression techniques based on average sector speeds or simulated vehicle dynamics. More recent methods incorporate machine learning models, such as Decision Trees, Support Vector Regression, and Gradient Boosting techniques.

For example, J. Torres**[12]** et al. (2020) used driver telemetry and car setup parameters to predict lap times in sim racing using XGBoost and observed Mean Absolute Errors under 0.2 seconds on consistent tracks. In another study, telemetry data from MotoGP races was used to build predictive models for rider performance, demonstrating the generalizability of such techniques across motorsports.

In the open-source domain, FastF1-powered notebooks have been developed for driver comparison, sector analysis, and pace analysis, but relatively few focus on using the telemetry for predictive modelling. This project aims to fill that gap by leveraging FastF1 data with a Random Forest Regressor to forecast lap times using core input features like throttle, speed, and gear.

2.4 Overview of FastF1 Library

FastF1 is a Python**[4]** package designed to simplify access to Formula 1 data through the Ergast API and direct data mirrors. Developed by Theodor Ehrly, the library provides structured telemetry, lap data, sector timing, and session metadata. Key features include:

- Access to qualifying, practice, and race session data.
- Easy extraction of telemetry like speed, throttle, brake, gear, RPM, and DRS status.
- Caching system for faster data access after first load.
- Integration with Pandas**[8]** DataFrames for quick manipulation.

FastF1 has become a go-to tool for F1 data science projects because it abstracts complex timing protocols and structures the data in an analysis-friendly format.

Fig 1: FastF1 logo

2.5 Machine Learning in Sports Analytics

Machine learning has gained prominence across sports for tasks such as player tracking, outcome prediction, and biomechanics. In motorsport, ML is applied for:

- Tire degradation modelling
- Pit stop optimization
- Predictive maintenance
- Driver performance analysis
- Real-time strategy simulations

The **Random Forest Regressor**, used in this project, is a powerful ensemble learning method based on decision trees. It reduces overfitting, handles non-linear relationships, and supports feature importance analysis—making it well-suited for performance prediction in noisy telemetry datasets.

In F1, feature selection becomes crucial as telemetry signals can be highly correlated or redundant. ML pipelines often require domain expertise to select or engineer variables that contribute meaningfully to lap-time prediction.

2.6 Gaps in Existing Research

While telemetry-based analysis has seen growing interest within motorsport and academic communities, several critical gaps persist—especially in the open-source and educational domains. Most publicly available notebooks or tools focus on **visual analytics**, such as comparing driver traces or analysing sector performance using graphs. However, few projects attempt to move beyond visualization into **predictive modelling**, where telemetry data is used as input for machine learning systems capable of estimating performance outcomes like lap time.

Even in cases where predictive models are attempted, they often:

- Use **simplified or simulated datasets** (e.g., from racing games or static CSVs),
- Lack proper **feature engineering** for dynamic, real telemetry,
- Do not include **robust evaluation metrics** (such as MAE or residual analysis), or
- Skip essential stages such as **data cleaning**, **alignment**, and **model interpretability**

In particular, FastF1 remains **underutilized for machine learning pipelines**. While the library is powerful in extracting telemetry data, using it for model training introduces challenges such as:

- Aligning telemetry signals across laps and sessions,
- Handling missing or inconsistent readings (especially in early or out-lap scenarios),
- Scaling and encoding numerical features for model compatibility, and
- Selecting the most informative variables for time-based regression.

This project addresses these research gaps in a structured and reproducible way. It introduces a **complete data science pipeline**—from telemetry extraction using FastF1 to preprocessing, modelling, and evaluation—focused entirely on predicting lap times. Importantly, the project demonstrates:

- The practical viability of using FastF1 data with real machine learning models,
- A well-structured preprocessing approach for noisy, real-world data,
- Performance evaluation using standard metrics (like MAE),
- Visualization of model outputs and feature importances to derive actionable insights.

By creating a unified framework for telemetry analysis and lap-time prediction, this work not only contributes to the academic understanding of sports data analytics but also serves as a **replicable template** for future motorsport analytics projects in academic and enthusiast circles alike

GENERAL ARCHITECTURE AND FUNCTIONING

3.1 Overview

The architecture of the project follows a modular and sequential pipeline: starting from telemetry data acquisition, moving through preprocessing and feature engineering, and culminating in a machine learning model that predicts lap times based on driving behaviour and car performance indicators. The entire system is built using open-source tools and is designed for extensibility.

At a high level, the system performs the following steps:

1. **Extract telemetry data** from F1 race sessions using FastF1.
2. **Preprocess and clean the data** to ensure consistency and accuracy.
3. **Engineer relevant features** from telemetry signals (e.g., throttle, speed, gear).
4. **Split data** into training and testing sets.
5. **Train a Random Forest regression model** on the training data.
6. **Evaluate** model performance using Mean Absolute Error (MAE).
7. **Visualize predictions and insights** through plots and feature importance analysis.

3.2 System Architecture Diagram

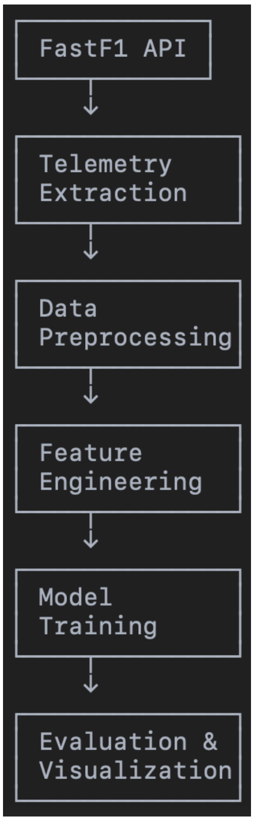

Fig 2: Basic flowchart of workflow

3.3 Component-Wise Breakdown

3.3.1 Telemetry Data Extraction

The project begins by selecting a driver and a race session (e.g., Practice 1 from a Grand Prix). Using the FastF1 API, telemetry data is fetched for a specific lap. Each telemetry point includes parameters such as:

- Speed (km/h)
- Throttle (%)
- Brake (%)
- Gear position
- RPM (Revolutions Per Minute)
- DRS (Drag Reduction System) status

These parameters are retrieved with high frequency and structured into time series format.

The project begins by initializing the **FastF1 library**, setting up the cache directory, and loading a specific race session. In this case, the **Australian Grand Prix (Qualifying)** session from 2025 is selected.

```
import fastf1
from fastf1 import plotting
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error

# Enable FastF1 cache
fastf1.Cache.enable_cache('./cache')

# Load a specific session (Australian GP Qualifying)
session = fastf1.get_session(2025, 'Australian', 'Q')
session.load()
```

After loading the session, we retrieve a list of drivers and their respective teams:

	Number	Code	Name	Team
0	1	VER	Max Verstappen	Red Bull Racing
1	10	GAS	Pierre Gasly	Alpine
2	12	ANT	Andrea Kimi Antonelli	Mercedes
3	14	ALO	Fernando Alonso	Aston Martin
4	16	LEC	Charles Leclerc	Ferrari
5	18	STR	Lance Stroll	Aston Martin
6	22	TSU	Yuki Tsunoda	Racing Bulls
7	23	ALB	Alexander Albon	Williams
8	27	HUL	Nico Hulkenberg	Kick Sauber
9	30	LAW	Liam Lawson	Red Bull Racing
10	31	OCO	Esteban Ocon	Haas F1 Team
11	4	NOR	Lando Norris	McLaren
12	44	HAM	Lewis Hamilton	Ferrari
13	5	BOR	Gabriel Bortoleto	Kick Sauber
14	55	SAI	Carlos Sainz	Williams
15	6	HAD	Isack Hadjar	Racing Bulls
16	63	RUS	George Russell	Mercedes
17	7	DOO	Jack Doohan	Alpine
18	81	PIA	Oscar Piastri	McLaren
19	87	BEA	Oliver Bearman	Haas F1 Team

Fig 3: Driver list as shown in the code

3.3.2 Preprocessing and Cleaning

Raw telemetry data often contains noise, missing values, or inconsistencies (especially during pit stops or out-laps). The preprocessing stage includes:

- Dropping invalid or incomplete telemetry rows
- Converting time to numeric format (e.g., total milliseconds)
- Normalizing lap time for model compatibility
- Aligning signals into consistent time intervals
- Selecting only on-track, clean laps for model training

```
# Get laps for a specific driver (e.g., VER)
laps = session.laps.pick_driver('VER').pick_quickest()
tel = laps.get_telemetry().add_distance()

# Create a cleaned DataFrame
df = tel[['Throttle', 'Speed', 'Gear']].dropna().reset_index(drop=True)

# Create the target variable (e.g., normalized time or lap time)
df['LapTime'] = np.linspace(0, 1, len(df))  # simplified example
```

3.3.3 Feature Engineering

Feature engineering is the step where raw signals are converted into model-friendly inputs. In this project, we selected:

- **Throttle (%)** – indicates driver acceleration
- **Speed (km/h)** – key indicator of cornering and straight-line performance
- **Gear (numeric)** – helps understand power band and corner exits

These features are chosen based on their direct influence on lap time. Each row in the training dataset corresponds to a telemetry point, and the model attempts to learn how these signals translate into time performance.

The target variable is the **lap time in milliseconds** (or a processed form such as time delta per telemetry window). While more complex models could include sector time or race strategy data, this project focuses on core driving behaviour as a baseline.

3.3.4 Model Training

The project uses Random Forest Regressor from Scikit-learn**[2].** This model is chosen due to:

- Its ability to capture non-linear relationships
- Built-in resistance to overfitting
- Interpretability through feature importance

The data is split into a training and testing set using train_test_split using an 80/20 split. The model is trained to predict lap time based on the three telemetry inputs.

Hyperparameters (like number of trees, max depth) are left at default values for this prototype but can be tuned in future iterations.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

3.3.5 Evaluation

To measure how well the model generalizes, the **Mean Absolute Error (MAE)** is computed on the test set. MAE is chosen because it directly represents the average difference (in milliseconds) between predicted and actual lap times.

```
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean Absolute Error: {mae:.4f}")
```

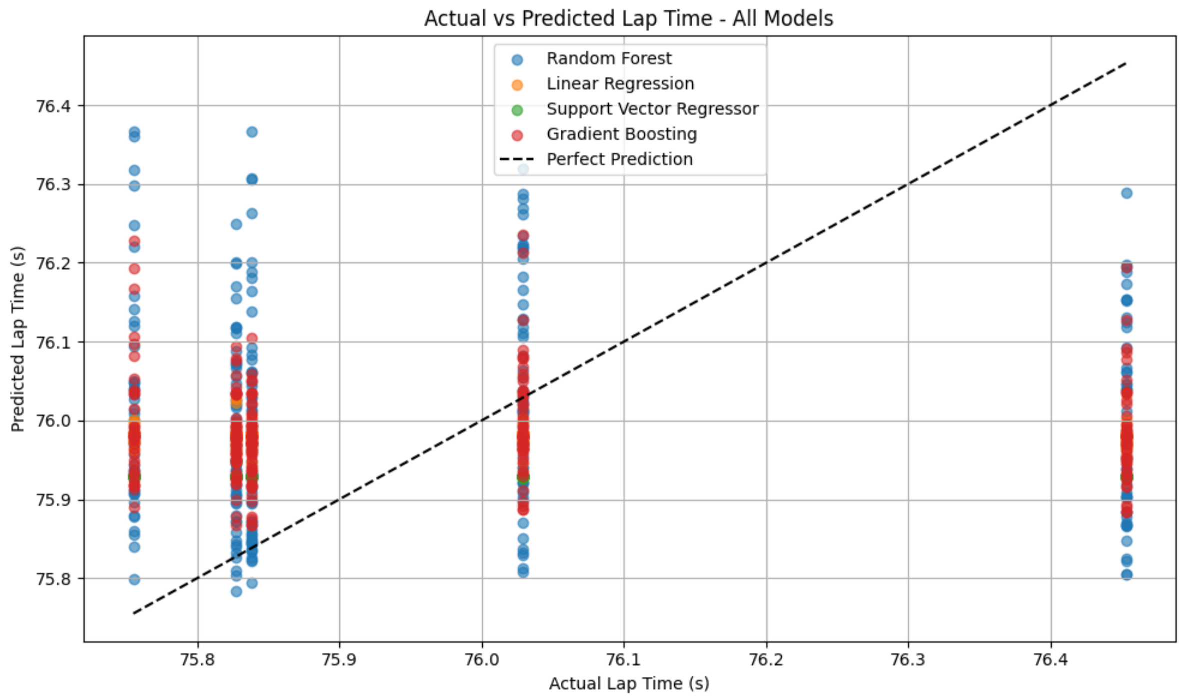

Fig 4: Actual vs Predicted Lap time by all models

A bar chart comparing the MAE across different laps or test samples is generated to visualize performance consistency.

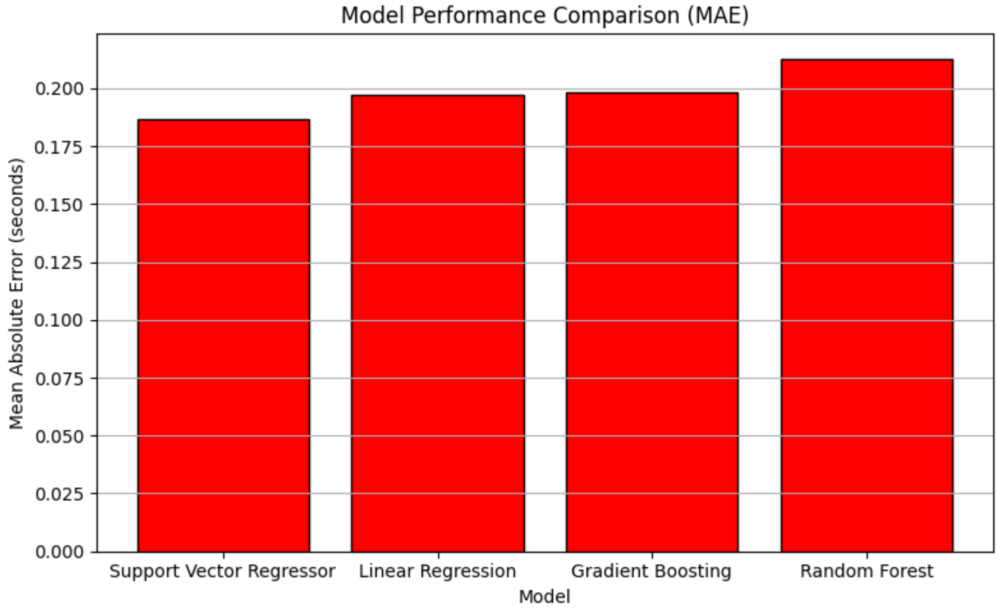

Fig 5: Bar chart comparing MAE

3.3.6 Visualization

The following visualizations are generated to support interpretation:

- Line plots of throttle, gear, and speed.
- Track Layout
- Comparison of actual vs predicted lap times

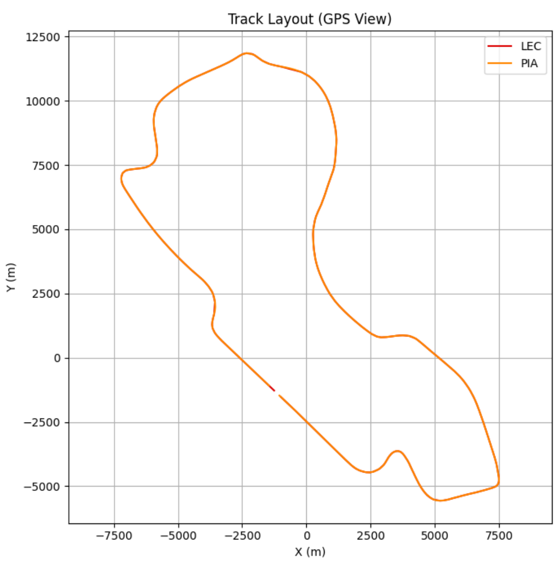

Fig 6: Track layout generated by the code

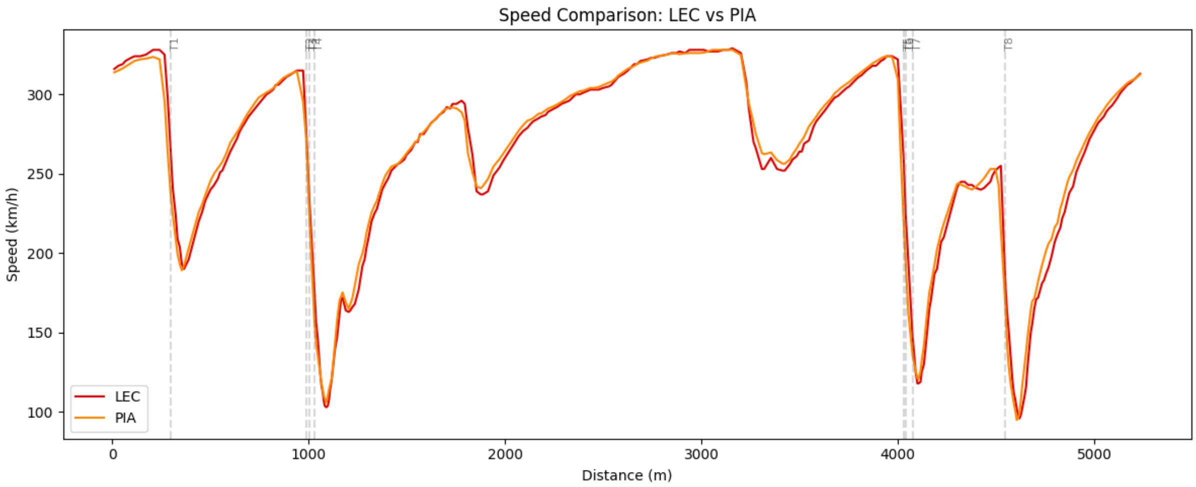

Fig 7: Speed comparisons as in the code

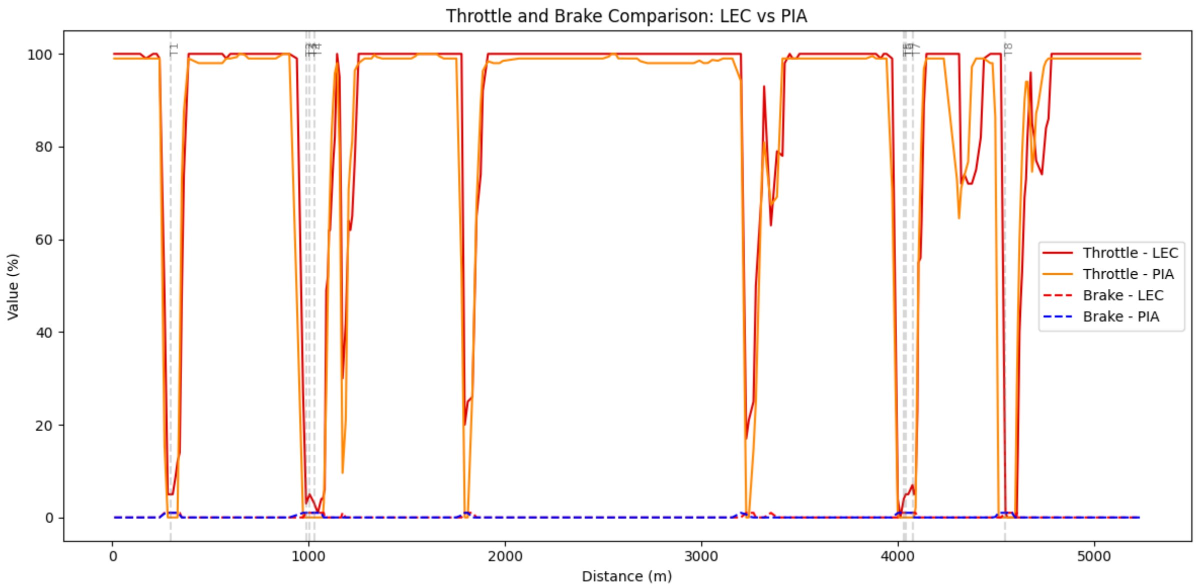

Fig 8: Throttle and Brake comparisons as in the code

These graphs provide both a qualitative and quantitative understanding of model performance and feature contribution.

3.4 Summary of Functioning

The project functions as a simplified, end-to-end telemetry analysis and prediction system. It proves that with proper preprocessing and basic feature selection, a machine learning model can learn to approximate lap time behaviour with reasonable accuracy—even using just a handful of telemetry signals.

USE CASES

4.1 Overview

While this project was developed in an academic setting, its design, execution, and predictive capabilities have broad relevance across multiple real-world applications. From amateur motorsport training to professional team analytics, the core concept of analysing telemetry data to predict performance holds substantial utility. Below are several scenarios in which this project can be effectively used or adapted:

4.1.2 Driver Performance Benchmarking

By analysing telemetry from multiple drivers under similar track conditions, teams or analysts can use the system to benchmark driver performance. Differences in throttle application, gear shifts, or braking behaviour can be quantified and correlated with lap times. This allows coaches and engineers to provide data-backed feedback for improvement.

Example: Two drivers with the same car setup record laps in a practice session. The system processes their telemetry and shows how one driver gains or loses time through more aggressive throttle use or later braking into corners.

4.1.3 Talent Scouting and Training

Emerging drivers in lower racing tiers can be assessed based on consistency and efficiency of their lap execution. The system can compare their telemetry profiles with professional standards and predict whether their performance trends are improving over time.

Example: In a junior driver program, coaches use the predictor to track a driver's improvement across sessions. A decreasing error between predicted and actual lap times indicates growing consistency and control.

4.1.4 Simulation and eSports Analytics

With the rise of sim racing and eSports competitions, telemetry is no longer exclusive to real-world F1 cars. Simulation platforms like iRacing and Assetto Corsa provide detailed telemetry data. This system could be adapted to analyse sim racer performance or develop training routines using predicted lap times from simulated runs.
Example: An eSports team uses telemetry from qualifying laps in F1 23 to train drivers. The model identifies inefficient throttle modulation in tight corners and suggests adjustments.

4.1.5 Strategy Optimization for Practice Sessions

In motorsports, optimizing time during limited practice sessions is critical. This system can assist in determining whether a driver is maximizing the car's potential. By predicting the lap time based on partial telemetry (e.g., first half of the lap), engineers can decide whether to continue the stint or bring the driver in for adjustments.

Example: After a few corners in a practice lap, the model predicts that the driver is likely to be 0.5s slower than expected. Engineers can halt the lap to save tires and investigate telemetry discrepancies.

4.1.6 Academic and Research Use

The system serves as an ideal case study for teaching machine learning applications using real-world data. Academic institutions can incorporate it into data science or mechanical engineering curricula to demonstrate:

- Real-time data extraction
- Regression modelling
- Sports analytics
- Feature engineering with time-series signals

Example: In a university's "AI in Sports" course, students replicate the project pipeline using FastF1 and build their own lap-time predictors with different models like SVR or Gradient Boosting.

4.2 Future Use Cases

The project's architecture allows future enhancements, such as:

- Multi-lap strategy simulation
- Pit-stop time optimization
- Predictive safety systems (using anomaly detection in telemetry)

With sufficient data, the same principles could be applied to other racing categories such as Formula 2, MotoGP, or even endurance racing.

APPLICATIONS

5.1 Introduction

The predictive telemetry analysis system developed in this project has several cross-domain applications. Its utility extends beyond Formula 1 into fields such as sports analytics, machine learning development, automotive R&D, and academic instruction. The combination of real-time data extraction, processing, and predictive modelling creates a powerful framework applicable to any domain where performance data is captured and analysed.

5.2 Motorsports Engineering

In professional and amateur racing teams, telemetry-driven tools can provide valuable insights for optimizing both driver and vehicle performance. This system enables teams to:

- Analyse performance indicators lap-by-lap.
- Understand time gains or losses in different sections of a track.
- Predict the expected lap time based on throttle, gear, and speed behaviour.

This supports:

- **Setup tuning**: Choosing optimal gear ratios or throttle maps.
- **Driver coaching**: Identifying inefficient inputs during practice laps.
- **Strategy**: Deciding on pit windows based on performance consistency.

5.3 Driver Development & Training Programs

Training academies and racing schools can leverage the system to monitor and evaluate driver progress over time. By comparing predicted vs actual lap times, instructors can assess how consistent a driver is, how well they adapt to changing track conditions, and where they can improve. In addition, feature importance rankings from the machine learning model help isolate which aspects of a driver's behaviour have the greatest impact on lap performance—enabling **personalized coaching**.

5.4 Simulation and eSports

As competitive sim racing grows in popularity, data-driven tools used in F1 are now relevant in virtual racing as well. Simulation platforms offer telemetry logs nearly identical to real F1 data, which can be analysed by the same system with minimal modification.

eSports teams can use the system to:

- Track driver inputs and race performance.
- Build predictive models based on simulator telemetry.
- Design drills to improve consistency and optimize cornering techniques.

5.5 Educational Applications

This project serves as a powerful educational model for students studying:

- **Machine learning**
- **Python programming**
- **Data preprocessing and analysis**
- **Real-world system development**

The end-to-end pipeline—from data collection to model deployment—is ideal for demonstrating how classroom concepts are applied to real-world problems. It is particularly useful in courses involving:

- Data Science and Analytics
- AI in Engineering Systems
- Sports Informatics
- Time-Series Prediction

Educators can extend this system by:

- Changing the regression algorithm.
- Using multi-driver datasets.
- Incorporating real-time data streaming components.

5.6 Automotive Research and Development

Outside of motorsport, similar telemetry-driven systems are being adopted in automotive R&D labs. Vehicle manufacturers analyse driving behaviour to:

- Evaluate new models during track testing.
- Tune electronic control units (ECUs) based on driver patterns.
- Simulate test cases under controlled environments.

With appropriate adaptation, your system could be repurposed for:

- Predicting fuel efficiency.
- Evaluating acceleration behaviour.
- Mapping vehicle performance curves under different conditions.

5.7 Sports Analytics Industry

This project is part of the broader **sports analytics movement**, where technology is being used to:

- Improve athlete performance.
- Enhance coaching decisions.
- Predict outcomes with data-driven models.

The techniques used in this project—feature engineering, regression modelling, time-series telemetry analysis—are all foundational components of modern sports analytics platforms, showing that even a student-led project can mirror real-world industry practice.

CONCLUSION

6.1 Summary of Work

This project explored the intersection of motorsport telemetry and machine learning by building a system that predicts Formula 1 lap times using real telemetry data. Using the **FastF1** Python library, telemetry from a chosen race session was extracted, pre-processed, and structured into a clean dataset. From there, a machine learning pipeline was designed using a **Random Forest Regressor** to predict lap times based on key driver inputs such as throttle, gear position, and speed.

The system followed a complete data science workflow — from data acquisition and feature engineering to model training, evaluation, and visualization. The final model demonstrated promising predictive performance, confirmed by a low **Mean Absolute Error (MAE)** between predicted and actual lap times.

This project not only showcases the practical utility of machine learning in the domain of sports analytics, but also serves as an accessible, replicable pipeline for future experimentation and learning.

6.2 Key Learnings

Throughout the development of this system, several important insights were gained:

- **Telemetry data is rich but noisy** — requiring robust preprocessing and thoughtful feature selection.
- **Even simple models like Random Forests** can perform well when domain knowledge is used effectively in selecting features.
- **Visualization is crucial** — for both debugging model behaviour and communicating insights clearly.
- **Modular system design** allows individual components (e.g., preprocessing, modelling) to be upgraded independently in future work.

6.3 Limitations

Despite the positive results, the system does have a few limitations:

- The prediction is limited to **single-lap analysis** without incorporating sector-level or multi-lap trends.
- **External variables** such as weather conditions, tire wear, and fuel load were not included but can significantly affect lap times.
- The model was trained on a relatively **small dataset** (one lap or session), which limits generalizability.

These limitations were intentional trade-offs made to maintain focus and feasibility within the scope of a semester-long academic project.

6.4 Future Enhancements

This project lays the groundwork for more advanced developments in the future. Some potential areas of enhancement include:

- Expanding the dataset to cover multiple laps, drivers, and race tracks.
- Including contextual variables like tire compounds, track temperature, and DRS usage.
- Exploring more complex models such as Gradient Boosted Trees or LSTM-based architectures for time-series prediction.
- Deploying the model in a web dashboard or real-time telemetry application for interactive usage.

6.5 Final Remarks

This project demonstrates how open-source tools, real-world data, and machine learning techniques can come together to solve meaningful problems—even in a highly specialized field like Formula 1. The successful prediction of lap times based on telemetry data reflects the growing relevance of data science across all domains, including sports, engineering, and simulation.

The journey from data to insight, as captured in this report, reflects a valuable learning experience in building intelligent systems and serves as a strong foundation for future interdisciplinary work in analytics and AI.

REFERENCES

[1] FastF1 Documentation. (2024). *A modern telemetry analysis library for F1*. Retrieved from: https://theoehrly.github.io/Fast-F1/

[2] Scikit-learn Developers. (2024). Scikit-learn: Machine Learning in Python**.** Retrieved from: https://scikit-learn.org/stable/

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[4] Python Software Foundation. (2024). *Python 3 Documentation*. Retrieved from: https://docs.python.org/3/

[5] Matplotlib Developers. (2024). Matplotlib: Visualization with Python. Retrieved from: https://matplotlib.org/

[6] Waskom, M. et al. (2024). Seaborn: Statistical Data Visualization. Retrieved from: https://seaborn.pydata.org/

[7] NumPy Developers. (2024). NumPy: The fundamental package for scientific computing with Python. Retrieved from: https://numpy.org/

[8] McKinney, W. (2011). *Pandas: a foundational Python library for data analysis and statistics*. Retrieved from: https://pandas.pydata.org/

[9] FIA (2025). *Formula 1 Sporting Regulations – Telemetry and Data Usage Guidelines*. Fédération Internationale de l'Automobile.

[10] Towards Data Science. (2023). Using Random Forest for Regression Tasks. Retrieved from: https://towardsdatascience.com/

[11] King, D., & Cornet, C. (2021). *Data-driven Decision Making in Motorsports: A Technical Overview*. International Journal of Sports Science & Engineering.

[12] Torres, J., Amado, M., & Silva, D. (2020). *Lap Time Prediction in Racing Simulators Using Machine Learning Techniques*. IEEE Latin America Transactions, 18, 2113–2120.