NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Big Data Analytics

Pitstop Predictions in Formula 1

Paulo Martins, r2015469

Beatriz Fonseca, r20201599

Eldar Medvedev, r20181162

Group 69

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

Project available on Github

Data available on OneDrive

INDEX

1. INTRODUCTION

With the rapid advancement of computing power and data acquisition technologies, **data analysis has become a cornerstone of competitive performance in Formula 1**. Teams now rely heavily on real-time and historical data, collected both on-track and in simulators, to monitor and optimize a vast array of metrics—from tire degradation and fuel consumption to driver behavior and weather conditions.

This data-centric approach allows teams to extract performance insights at a granular level, enabling them to make informed decisions that can be the difference between victory and defeat. One of the most critical strategic elements in a race is **pit stop timing**, which can be influenced by numerous dynamic factors including track position, tire wear, weather conditions, and race incidents.

The primary motivation for this project is to leverage **lap-by-lap race data, car telemetry, and weather variables** to build a predictive model that can forecast whether a driver is likely to pit on the next lap. Such a tool could have valuable applications in race strategy development, simulation modeling, and even broadcasting insights, providing a competitive edge in a sport where milliseconds matter.

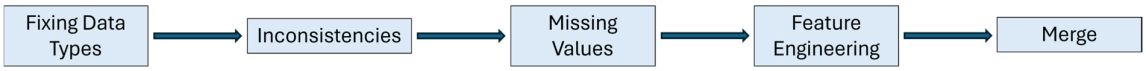
2. DATA COLLECTION

The dataset used in this project is structured and encompasses three primary domains: **lap-by-lap data, car telemetry, and weather conditions**. This data is sourced from the **FastF1 Python library** developed by *T. Oehrly (2019)*, which provides access to detailed Formula 1 race information derived from official sources.

The data from this library is provided as **extended Pandas DataFrames.** Therefore, an auxiliary script, TK, was executed in a local environment to extract the relevant data and then export it to CSV format. This process facilitates seamless integration with the Databricks analytics platform, where further processing and model development were conducted.

3. DATA PREPROCESSING

To prepare the dataset for the modeling stages, a series of preprocessing steps were undertaken, as illustrated in the figure below.



Initially, for each dataset segment—lap data, telemetry, and weather—features that were considered **irrelevant** to the prediction task were dropped, and all remaining feature **data types** were carefully verified and corrected where necessary.

Next, the dataset was examined for **consistency issues**. Laps that had missing values in all three sector time features (*Sector1Time*, *Sector2Time*, and *Sector3Time*) were removed, as they provided no meaningful performance information. In such cases, the previous lap was flagged as a DNF (Did Not Finish), acknowledging the likely early termination of their race.

Following this, the focus shifted to handling **missing values** across several key features, including *LapTime*, *PitOutTime*, *PitInTime*, the three sector times, and four speed-related metrics: *SpeedI1*, *SpeedI2*, *SpeedFL*, and *SpeedST*. Missing values in *PitOutTime*, *PitInTime*, and the sector time features were left untouched, as they were not intended for direct use in model training but rather as potential sources for engineered features. In contrast, missing *LapTime* values were logically reconstructed by calculating the difference between the lap's end time (*LapSessionTime*) and start time (*LapStartTime*).

For the speed features, a hierarchical imputation strategy was applied. The first approach involved filling missing values with a rolling average calculated from the driver's previous laps, ensuring no future data was used in the process. If this method failed due to a lack of sufficient prior data, the missing value was imputed using the teammate's speed on the same lap, leveraging the assumption of similar car performance. Lastly, remaining gaps – registered only on *SpeedFL* - were filled using the value from *SpeedST*, based on the observation that the finish line often aligns with the circuit's longest straight.

Lastly, **feature engineering** was carried out with the goal of creating more informative and predictive variables to enhance model performance. Given the time-series nature of the dataset, the *Window* function was used and abused to generate features that captured temporal patterns and contextual relationships across laps. This function enabled the calculation of rolling statistics and lagged values, which capture driver behavior and performance trends over time. Without leveraging this function, the project would have faced significant challenges, particularly in terms of computational efficiency and infrastructure limitations, as alternative approaches would have required considerably more resources for data transformation and storage.

As the final step in the preprocessing pipeline, the telemetry and weather datasets were **aggregated** at the lap level to match the granularity of the lap data. Once aligned, all three datasets were seamlessly **merged** into a single, unified dataset, forming a comprehensive view of each lap that combines driver performance, car behavior, and weather conditions—laying the foundation for the modeling stage that follows.

4. METHODOLOGY

The modeling stage began with the creation of the **target variable** *WillPitNextLap*. This was derived using lagged values of the *PitInTime* feature: if a value was recorded for *PitInTime* on a given lap, the preceding lap was labeled with a '1' in the target variable, indicating that a pit stop would occur on the following lap. All other laps were assigned a value of '0', signifying no pit stop in the next lap.

Once the target variable was defined, the dataset was **split into training and testing subsets**, and categorical features were processed using a combination of *StringIndexer* and OneHotEncoder. These transformations were then incorporated into a pipeline to ensure consistency and reusability.

To train the model, a **sequential cross-validation** loop was implemented. For each lap from lap 5 up to the maximum lap, the model was trained using all data up to and including the current lap and then evaluated on the immediate next lap. This setup simulates a real-time prediction environment, where

only historical data is available at prediction time. Importantly, this approach preserves the temporal order of the data, thereby preventing any leakage of future information into the training process.

A *RandomForestClassifier* was employed as the predictive model, and performance was evaluated for each iteration using metrics such as the area under the precision-recall curve (AUPRC) and Receiver Operating Characteristic Curve (AUC-ROC), along with F1-score, precision, recall, and accuracy.

After the evaluation loop, the model was retrained using all data from the training set and predictions were made on the test set. At the end, a confusion matrix was generated to visualize the model's predictions.

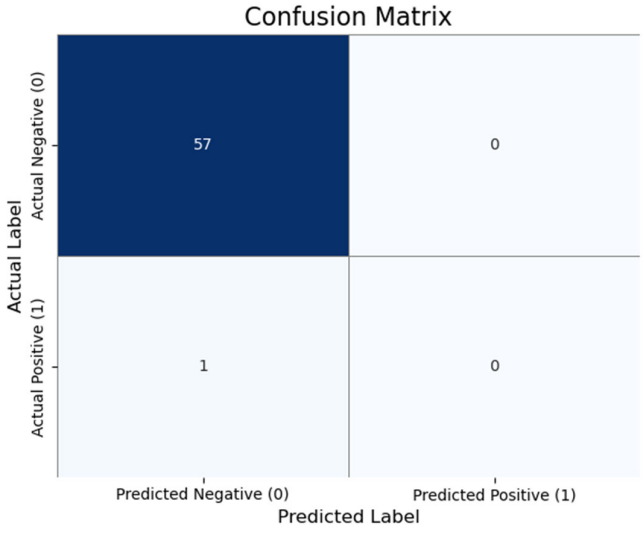
5. RESULTS

The final model produced the following results:

Accuracy	F1	Precision	Recall	AUC-ROC	AUPR
0.9828	0.9742	0.9658	0.9828	0.5000	0.0172

While the first four metrics - accuracy, F1 score, precision, and recall - appear to suggest strong performance, they are misleading in this context due to the **severe class imbalance** in the dataset. AUC-ROC and AUPR offer a more truthful reflection of the model's ability to generalize. The AUC-ROC score of **0.5000** indicates that the model performs no better than random guessing when it comes to distinguishing between pit and non-pit laps. Even more concerning, the **AUPR of 0.0172** confirms that the model struggles to make meaningful predictions, as it fails to achieve a reasonable balance between precision and recall in the presence of rare positive cases.

This issue is further highlighted by the **confusion matrix**, which reveals that the model has likely defaulted to always predicting the majority class - laps where no pit stop occurs. In doing so, the model avoids false positives but misses nearly all actual pit stops, leading to poor real-world utility despite superficially high scores on some metrics.



6. CONCLUSION

While the results of our project were, by most objective standards, disappointing, there is no denying the core challenge was meaningfully explored and that significant progress was made in understanding the complexities of the problem. Notably, a model was successfully trained using machine learning techniques on telemetry data - a demanding data type both in terms of size and structure. More importantly, the work lays the foundation for future research in a highly promising area.

Initially, the problem was approached using standard cross-validation methods along with off-the-shelf algorithms like gradient boosting and decision trees. This proved to be a poor fit. A deeper analysis revealed that the problem involved multi-variable time series panel data, also known as longitudinal data. Complicating matters further, the data set qualified as big data: a single training run for one driver, in one race, could exceed five hours. Our resource limitations made extensive experimentation impractical.

This realization led to a key methodological breakthrough. Traditional cross-validation strategies failed because they didn't respect the temporal structure and dependencies in the data. Going forward, a more suitable model architecture is needed - ideally one capable of sequential learning. Attention-based models, Markov chains, or hybrid approaches might better capture the temporal, state-dependent nature of the data. Such models could account for race- and driver-specific behavior while incorporating important predictive features like tarmac temperature and tire selection.

A second major challenge is the design of an appropriate loss function. The standard losses available in common frameworks were ineffective. On inspection, this appears tied to the data distribution: as time steps become smaller, observable events like pit stops become increasingly rare. This raises a key modelling question - how do we accurately detect and learn from "moments of decision" in a dense, continuous stream of telemetry?

In summary, the disappointing performance of our models is not surprising in retrospect. We chose an exceptionally complex problem - both technically and computationally - without realizing the full extent of its difficulty at the outset. However, we do not view this as a failure. Rather, we believe our work has uncovered fertile ground for future exploration. The insights gained, especially around model selection and validation strategy, are meaningful and will inform more effective approaches in subsequent efforts.

7. REFERENCES

Oehrly, T. (2019). *FastF1 documentation*. Retrieved May 10, 2025, from https://docs.fastf1.dev/

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://OTexts.com/fpp3/