# Formula 1 Race Data Analysis: Tyre Degradation and UnderCut Strategy - A Detailed Project Report

Jeet Prajapati

June 25, 2025

**Abstract**

This report details the development of a multi-stage machine learning pipeline designed to analyze and predict Formula 1 race strategy. Utilizing publicly available Kaggle data, the project first constructs a sophisticated Pace & Degradation model using an XGBoost Regressor to predict lap times as a function of tire age and other variables. This regression model achieves an $R^2$ of 0.701, effectively quantifying the impact of tire wear. The predictions from this model are then used as inputs for a second, "When to Pit?" classification model. This Random Forest classifier, trained to handle imbalanced data, demonstrates high efficacy with a precision of 0.94 and recall of 0.89 in identifying optimal pit stop laps. The project culminates in two expansions: a hyperparameter tuning process using `RandomizedSearchCV` with a `TimeSeriesSplit`, and the creation of a practical, high-level "Undercut Advantage Calculator." This tool successfully simulates a historical race scenario, quantifying a potential 1.350-second strategic advantage, thereby validating the end-to-end potential of a data-driven approach to F1 strategy.

## 1 Introduction

The competitive landscape of Formula 1 is defined by fractions of a second, where on-track performance is inextricably linked to strategic decision-making. This project moves beyond simple historical analysis to build a predictive toolkit that can help understand and forecast key race events. By leveraging a comprehensive dataset encompassing race results, lap times, and pit stops, we aim to model complex phenomena like tire degradation and pit stop timing.

The project is executed in a sequential, two-part modeling process, where the output of the first model becomes a critical feature for the second. This is followed by expansions to refine the initial model and to build a tangible strategic tool.

- **Part 1: Pace & Degradation Model:** A regression model to predict lap times, which serves as a proxy to model the effects of tire degradation.
- **Part 2: "When to Pit?" Model:** A classification model that uses the pace model to predict the optimal moment for a pit stop.
- **Expansions:** We will then improve our pace model with hyperparameter tuning and build a strategic "Undercut Advantage Calculator".

The technical stack employed includes `pandas`, `scikit-learn`, `XGBoost`, `matplotlib`, and `seaborn`.

## 2 Data Acquisition and Preparation

### 2.1 Data Sources

The project utilizes four key datasets sourced from Kaggle:

- `races.csv`

- `results.csv`
- `lap_times.csv`
- `pit_stops.csv`

### 2.2 Data Merging and Cleaning

The raw data was merged into a single comprehensive DataFrame. The process began with the `lap_times` data, which was progressively enriched by merging with `races` (on `raceId`), and then `results` (on `raceId` and `driverId`).

To ensure data quality and relevance, several cleaning steps were performed:

- The first lap of each race was removed (`lap > 1`) as it is an outlier due to the standing start.
- Lap times were converted from milliseconds to seconds.
- A critical outlier removal technique was implemented: for each race, the median lap time was calculated. Any lap time that was 15% greater than the race's median was discarded. This effectively removes laps affected by safety cars, accidents, or significant errors, ensuring the model learns from representative race pace.

### 2.3 Feature Engineering for Pace Modeling

The core of the degradation model relies on quantifying the age of the tires. The following key features were engineered:

- `stint`: A numerical indicator for each continuous period of racing between pit stops.
- `LapInStint`: This crucial feature counts the number of laps a driver has completed on a specific set of tires (i.e., within the current stint). This serves as our primary proxy for tire age and wear. It was created by grouping the data by race, driver, and stint, and then applying a cumulative count.

## 3 Methodology Part 1: Pace and Degradation Regression Model

### 3.1 Model Selection

An **XGBoost Regressor** (`xgb.XGBRegressor`) was chosen for its proven performance on structured, tabular data. The model's objective was set to `reg:squarederror` and its performance was evaluated using the Mean Absolute Error (`mae`). Initial hyperparameters were manually selected, including `n_estimators=1000` and `learning_rate=0.05`.

### 3.2 Chronological Training and Testing

To simulate a realistic forecasting scenario, a chronological data split was enforced.

- **Training Set:** All data from years **before 2024**.
- **Test Set:** All data from **2024 onwards**.

This ensures the model is never tested on data that occurred before its training data, preventing data leakage and providing a more robust evaluation of its predictive power. Categorical features (`constructorId`, `circuitId`) were one-hot encoded using `pd.get_dummies`.

### 3.3 Performance Evaluation

The trained model was evaluated on the 2024 test set, yielding the following results:

- **Mean Absolute Error (MAE): 3.929 seconds**. This means, on average, the model's lap time prediction was off by just under 4 seconds. While this indicates room for improvement, it's a strong result given the lack of specific tire compound data.
- **R-squared ($R^2$): 0.701**. This is a strong result, indicating that the model could explain over 70% of the variance in lap times.

Visual analysis confirmed the model's effectiveness. A scatter plot of actual vs. predicted times showed a strong positive correlation, and a line plot comparing the average actual and predicted lap times against `LapInStint` showed that the model accurately captured the upward trend in lap times as tires aged, thus successfully modeling degradation.

## 4 Methodology Part 2: "When to Pit?" Classification Model

### 4.1 Problem Formulation and Target Variable

This model addresses the question of *when* a driver should pit. It is a binary classifier that predicts for any given lap whether a pit stop is the optimal decision (`should_pit = 1`) or not (`should_pit = 0`). The target variable was created by flagging laps in the main DataFrame that corresponded to a recorded pit stop in the `pit_stops.csv` file.

### 4.2 Predictive Feature Engineering

The key innovation of this model is its use of the Pace Model to simulate future scenarios and create predictive features.

1. **Current Pace Prediction**: The Pace Model was used to predict the `predicted_current_lap_time` for every lap in the dataset.
2. **"No Pit" Scenario**: For each lap, a hypothetical "next lap without pitting" was simulated by incrementing `LapInStint` by 1. The Pace Model then predicted the lap time for this scenario (`predicted_next_lap_time_NO_PIT`).
3. **"Pit" Scenario**: A second hypothetical "next lap with a pit stop" was simulated. This involved incrementing the `stint` number and resetting `LapInStint` to 1 (representing fresh tires). The Pace Model predicted the lap time for this fresh-tire scenario (`predicted_next_lap_time_WITH_PIT`).
4. **Pit Advantage Calculation**: The core predictive feature, `pit_advantage_seconds`, was calculated using the formula:
   `pit_adv_s = (pred_next_lap_NO_PIT) - (pred_next_lap_WITH_PIT + TIME_LOST_IN_PITS)`
   where `TIME_LOST_IN_PITS` was set to a constant 22 seconds. A positive value indicates a time advantage to be gained by pitting.

### 4.3 Model Training and Imbalance Handling

A `RandomForestClassifier` was chosen for this task. As pit stops are rare events, the dataset is highly imbalanced. To counteract this, the `class_weight='balanced'` parameter was used, which adjusts the model's training process to give more weight to the minority (pit stop) class.

### 4.4 Performance Evaluation

The strategy model performed exceptionally well on the test set:

- **Precision (for class 1): 0.94**. Of all the laps the model predicted as pit stops, 94% were correct.
- **Recall (for class 1): 0.89**. The model successfully identified 89% of all actual pit stops.
- **F1-Score (for class 1): 0.92**. The harmonic mean of precision and recall indicates a highly robust model.

The confusion matrix visually confirmed these results, showing a very low number of false positives and false negatives, making it a highly reliable decision-support tool.

## 5 Project Expansions

### 5.1 Pace Model Hyperparameter Tuning

To ensure the Pace Model was optimally configured, `RandomizedSearchCV` was employed. A `TimeSeriesSplit` with 3 splits was used for cross-validation to preserve the chronological order of the data. The search explored 25 different combinations of hyperparameters. The best combination yielded an MAE of **5.244 seconds**. Interestingly, this was higher than the manually selected parameters' MAE of 3.929 seconds, suggesting the initial parameters were already near-optimal and that the model's performance is likely limited more by the available features than by its configuration. The best hyperparameters found were: {`'subsample': 0.7, 'n_estimators': 500, 'max_depth': 4, 'learning_rate': 0.05, 'colsample_bytree': 0.8`}.

### 5.2 Strategic Application: The Undercut Advantage Calculator

A practical function, `calculate_undercut_advantage`, was built to leverage the Pace Model for strategic simulation. The "undercut" involves pitting one lap before a rival to gain an advantage through the combination of a fast out-lap on new tires versus the rival's slower in-lap on old tires.

The function simulates a two-lap window:

- **Your Car**: Pits on the current lap (N), incurring pit loss, then completes lap N+1 on 1-lap-old tires.
- **Rival Car**: Stays out on lap N, then pits on lap N+1, incurring pit loss.

The total time for both drivers over these two laps is calculated, and the difference reveals the advantage. When applied to the 2021 Bahrain GP (`race_id=1052`), simulating Hamilton (`driver_id=1`) undercutting Verstappen (`driver_id=854`) on **lap 12**, the calculator predicted a **net advantage of 1.350 seconds** for Hamilton.

## 6 Conclusion

This project successfully demonstrated the viability of a multi-layered, data-driven approach to F1 strategy analysis. By first modeling a fundamental aspect of racing—pace and tire degradation—it was possible to build a powerful and accurate decision-making tool for a complex strategic choice like pit stops. The final Undercut Calculator provides a tangible example of how these predictive models can be transformed into actionable strategic insights, offering a significant proof-of-concept for the application of machine learning in motorsport.