Neural Network Driven Pit-Stop Strategy in Formula One

Kurtis Wiles

ID: 009469654

Western Governors University

C964 Capstone Project

October 18, 2024

Table of Contents

LETTER OF TRANSMITTAL

October 18, 2024

Mr. G
Generic Refined Motor Company
921 Shoebox Ave.
Raleigh, North Carolina

Dear Mr. G,

GREM Co.'s debut in Formula One has exceeded everyone's expectations. Our race team has consistently performed on par with some of the best in the sport, and I am confident that our first race victory is within reach. However, our race strategy—particularly our pit stop strategy—has not matched the level of our competitors. Despite the speed of our cars and the skill of our team, we have struggled to position our drivers for race wins. The primary challenge lies in the relative inexperience of our race engineers compared to other teams, and this gap is holding back our potential.

To overcome this, I propose developing a machine learning driven application to analyze real-time race data. This application would support our race engineers in making more informed strategic decisions, ultimately giving our drivers a competitive edge. By training a neural network on successful pit-stop strategies from the past, the system could help identify optimal pit-stop windows in real-time, ensuring we capitalize on every opportunity to outmaneuver our rivals.

The proposed project to develop this application will have an upfront cost of $5,100, with labor estimated at $12,500. After deployment, the application will require weekly updates with the latest race data to maintain competitive performance, costing an estimated $800 in labor for each active week of the season. Once approved, the project will take over a month to complete and deploy, allowing us to test it in a real race before the end of the 2024 season. This also provides time for any necessary refinements before the start of the 2025 season.

As a machine learning specialist with three years of experience developing neural network driven applications, I am confident that this project will significantly improve our race engineers' decision-making and help bring us closer to our first race win.

Thank you for considering my proposal. I look forward to your response and the opportunity to begin work on this application.

Sincerely,

Kurtis Wiles

Kurtis Wiles

PROJECT PROPOSAL PLAN

Project Summary

Background

Formula One, sanctioned by the FIA, has become the world's most popular international racing series. It is a multibillion-dollar enterprise with a rich history spanning decades. Manufacturers like GREM Co. who participate as constructors develop cutting-edge technology for their race cars, aiming to secure championships for their drivers and teams. While winning races certainly depends on each car's performance and the driver's skill, race strategy—in particular, pit stop strategy—is equally crucial. A car cannot complete a race on a single set of tires, requiring pit stops to change them throughout the event. The timing of these pit stops can often mean the difference between winning and losing a race.

Several factors influence the decision to make a pit stop, including track conditions, tire wear, available tire compounds, the position of teammates on the track, and the strategies of rival teams. While a race strategy can be planned, teams must remain flexible and adapt to the unpredictable, ever-changing dynamics of an ongoing race to stay competitive.

Problem

As the newest constructor in Formula One, GREM Co. has quickly established itself by performing on par with some of the sport's most renowned teams. This success is due in part to the cutting-edge research and technology behind the development of our cars, as well as the dedication of our drivers and team members who consistently perform at their best. However, despite this impressive debut, it is evident that our team's strategic decisions—particularly regarding pit stops—leave room for significant improvement.

GREM Co.'s pit stop strategy has been inconsistent and has failed to put our drivers in the best position to capitalize on their potential. As a result, we have consistently lost track position to teams that outmaneuver us purely on strategy. While we have the speed and the capability to win races, our pit stop strategy is not yet at a championship-winning level.

Our race engineers often find themselves reacting to the strategies of more experienced teams. We lack the depth of expertise that others have developed over decades in the sport. To fully unlock our potential, GREM Co. must find a solution that enables our race engineers to make more effective pit stop decisions during races.

Solution

The solution to GREM Co.'s strategic needs lies in a machine learning application designed to assist race engineers with real-time pit stop decisions. By analyzing live race data, this system will help to identify the optimal timing for a car to make a pit stop, maximizing race performance. A neural network, trained on historical data from successful strategies, will recognize key indicators of an ideal pit stop window and apply this knowledge to new, evolving race conditions in real-time.

This application will include a trained neural network model capable of replicating successful pit-stop strategies. It will process race data lap by lap, providing an output that indicates the model's confidence in whether the next lap is an optimal time for a pit stop. The application will feature a user-friendly interface for manual data input and the potential for future integration with our existing systems for automated data input. A user guide will also be provided, detailing installation and basic usage.

Data Summary

Data Processing

The data that will be used to train this neural network comes from Kaggle (Rao, 2024) and originates from the Ergast Motor Racing Developer API (Newell). It contains comprehensive lap-by-lap data from nearly every Formula 1 race dating back to 1950. However, before this data can be used for training, it will require significant preprocessing. While it may seem advantageous to train the model using all 500,000+ laps of data available, it is important that the model only learns from relevant data. To ensure the best outcome, we will filter out and use only the most successful and recent strategies before the development of the application.

Data from races before 2014 will be removed, as this marks the beginning of Formula 1's 'hybrid era,' making earlier strategies less relevant to today's competition. Additionally, data from cars that finished in a worse position than their starting grid position will be removed, as this generally indicates a failed strategy or otherwise bad race outcome. We will also exclude data from cars that finished outside the top 10 to focus the model on more successful strategies. Further analysis will help identify and eliminate any additional noisy or irrelevant data, ensuring that only the most effective strategies remain for training the model.

Data Use

The remaining data after processing will effectively meet the needs of this project. It represents the type of data points that are relevant and accessible in a real-time race setting. These data points will help the network identify when a car is performing suboptimally and in need of fresh tires, as well as when rival teams are making strategic moves to try and gain an advantage.

After the development and deployment of the completed application, the data will not be actively used; however, it will be necessary to keep this data and continuously add new race data as it becomes available. Regular retraining of the model with this updated data will ensure that the application remains current and continues to provide the most competitive pit stop strategies. This process will help keep our system adaptable to evolving race strategies as the cars continue to evolve and improve with every race.

There are no ethical or legal concerns regarding the data used in this project, as all information is publicly available and accessible for use.

Implementation

Methodology

The development of the application will follow the SEMMA methodology (Sample, Explore, Modify, Model, and Assess). This methodology emphasizes an iterative approach focused on continuous improvement and model building, specifically for data mining and analysis. SEMMA is well-suited for our project because it mirrors the exact steps required for developing our application and training the neural network, including data sampling, exploration, modification, and model construction. Each phase of SEMMA will guide development in refining both the dataset and the model to achieve the best results.

Outline

Sample: Historical race data is available dating back to the 1950s. We will focus on sampling data from 2014 to the present, as this period represents the most relevant information from the current 'hybrid era' of Formula 1. Additionally, three races from the 2024 season will be separated and reserved for testing purposes.

Explore: Exploratory data analysis will be conducted to identify key data points that can provide insights into successful pit stop strategies. This phase will also involve detecting outliers and bad data, which will need to be removed to ensure that the neural network is trained on data from the most successful strategies.

Modify: Additional useful metrics will be calculated from the data, such as lap time splits, time gaps between cars, and the pit stop decisions of other cars on the track. These additional insights will provide the model with more factors to consider when training and making pit-stop decisions. The data must also be normalized to ensure consistency during training and real-time data input after deployment.

Model: The processed dataset will be used to train a neural network. For each lap, the model will receive inputs that we identify as key factors influencing pit-stop decisions, such as lap times, gaps to other cars, and pit-stop decisions by other teams. It will also receive a Boolean value indicating whether a pit stop occurred on that lap or if the car stayed out. Through this, the algorithm will learn the optimal conditions that signal when a pit stop is most beneficial.

Assess: The resulting machine learning model will be evaluated using the data that was set aside from the training set for testing. Our goal is to achieve a recall score of 70% or higher, ensuring the model can reliably identify optimal pit stop windows. If the model does not meet this benchmark, we will revisit our data processing, feature selection, and neural network architecture to make improvements and enhance performance.

Timeline

This project will require approximately 125 work hours to complete, spread over a little more than one month. A timeline of key project milestones is outlined below.

Milestone	Duration	Start	End
Project Approval	3 hours	10/21	10/22
Collect and Sample Data	10 hours	10/23	10/25
Data Preprocessing	35 hours	10/26	11/1
Develop Neural Network	10 hours	11/2	11/5
Train the Network	25 hours	11/6	11/13
Validate the Model	15 hours	11/14	11/18
Develop GUI	10 hours	11/19	11/21
Application Deployment	10 hours	11/22	11/25
Post Implementation Report	7 hours	11/26	11/28

Evaluation Plan

Verification in Development

During data processing, it will be crucial to ensure the quality of the data used for training the neural network. This will involve removing strategies that fall outside normal parameters, such as pit stops that occur unusually early or late in a race, as these typically represent exceptional circumstances. Additionally, data will be inspected using histograms and scatterplots to identify and eliminate potential outliers, ensuring the neural network isn't trained on skewed data.

Once the data has been cleaned and processed, attention will shift to training the model. Throughout the training process, the network's learning effectiveness will be monitored by plotting the loss function over time. This will allow us to confirm that the network is reducing its loss on the training data while avoiding overfitting.

Final Validation

Upon completion of the project, the effectiveness of our model will be tested against data from three races in the 2024 season of Formula One. This will test its effectiveness and relevancy in current-day competition. The model will run on the test data and have its pit stop predictions scored against the actual pit stop decisions made by teams during the races. Results will be scored by the binary classification metrics of precision, recall, and F1-score (not to be confused with Formula One). If the model can achieve a recall of 70% or better, it will demonstrate its effectiveness, value to our race team, and ability to offer crucial insights to our race engineers.

Resources and Costs

Hardware and Software

Resource	Description	Cost
Workstations	Computers for each of the project team members.	3 x $1,200
Nvidia GPU	GPU with CUDA cores for efficient training of the neural network	$1,500
Software	Windows 11 and the Python distribution Anaconda are both free to use	Free
TOTAL:		**$5,100**

Labor Costs

Employee	Time	Rate	Cost
Data Analyst	45 hours	$100	$4,500
ML Engineer	60 hours	$100	$6,000
UI/Software Developer	20 hours	$100	$2,000
TOTAL:	**125 hours**		**$12,500**

Maintenance

Collecting and adding data from new races as well as retraining the neural network will require around 8 hours of labor every week of the active season.

Maintenance	Time	Frequency	Rate	Cost
Race Data Collection	6 hours	Every Race (Weekly)	$100	$600
Retraining Neural Network	2 hours	Every Race (Weekly)	$100	$200
TOTAL:	**4 hours**	**Weekly**		**$800**

POST-IMPLEMENTATION REPORT

Solution Summary

Purpose

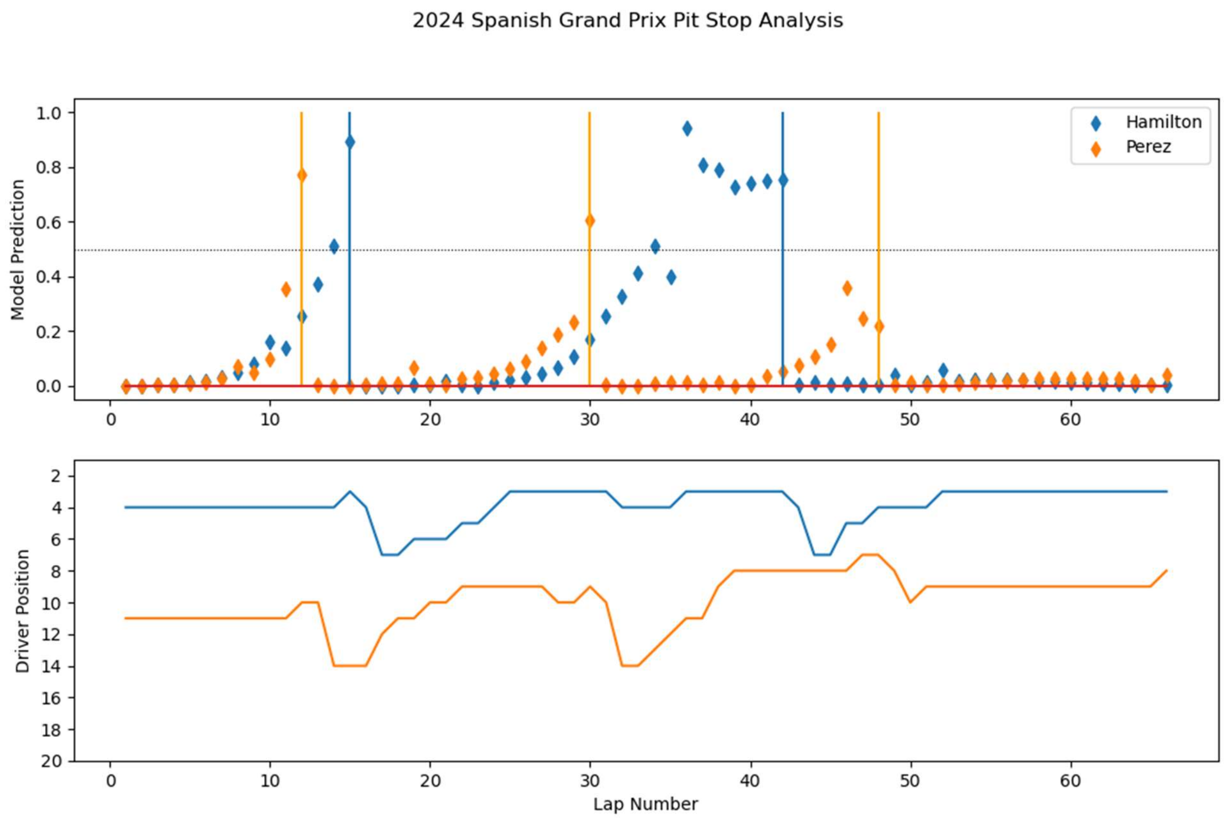
GREM Co.'s race team has shown impressive speed and efficiency in our debut year as the newest Formula One constructor. However, despite our potential, our performance has been consistently undermined by poor pit stop strategy. Teams with significantly more experience have been outperforming us in this area, preventing us from putting our drivers in a position to win races. This project has aimed to solve GREM Co.'s pit stop strategy needs by developing a machine learning driven application to assist race engineers in making real-time pit stop decisions.

With the completion of this project, we have developed a neural network model capable of analyzing live race data to help identify the optimal timing for pit stops. Trained on historical data from successful strategies, the model can recognize key indicators of an ideal pit stop window in new, unseen races. By processing race data lap-by-lap, the model provides a confidence output that indicates whether the next lap presents an optimal opportunity for a pit stop. This application can now be used by our race engineers during a race to help make more competitive pit stop decisions and position our drivers to win.

Results

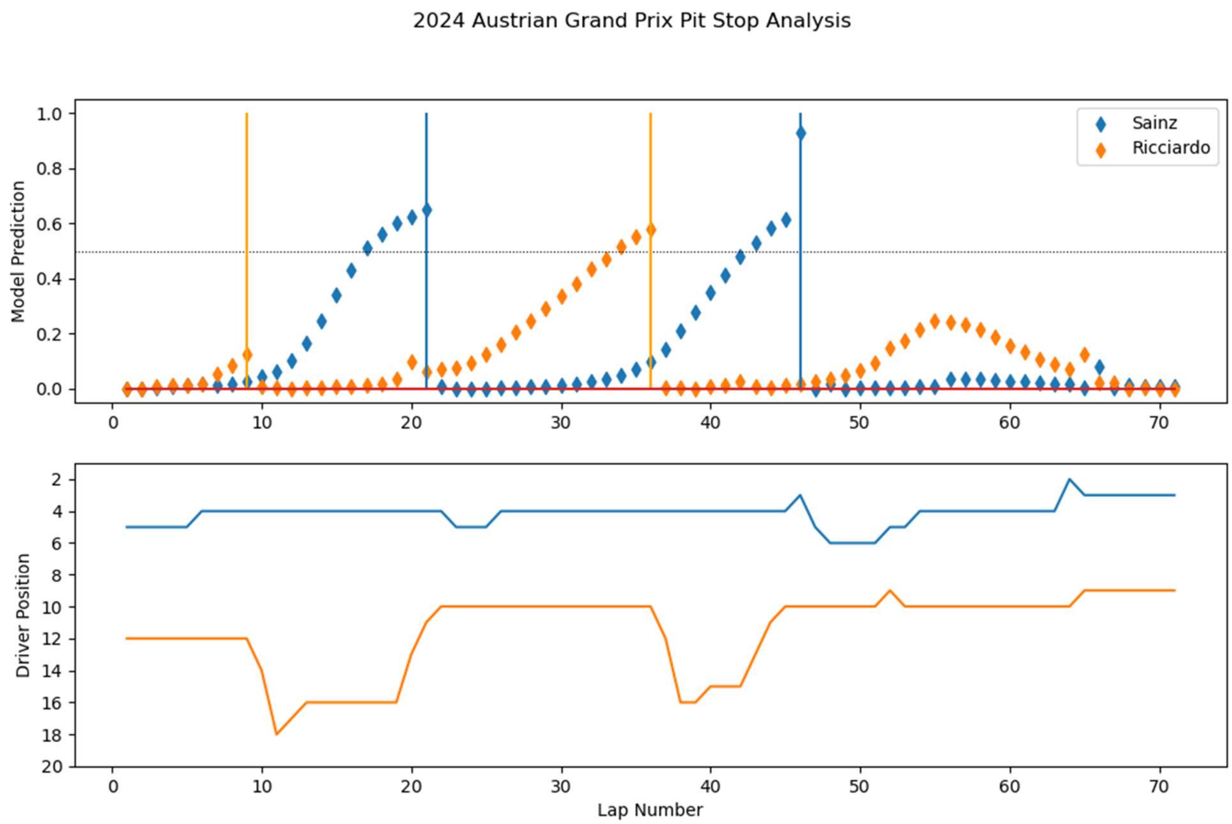
The model performs well on our test data, providing insightful and actionable information. To fully appreciate the model's effectiveness, it is helpful to visualize its outputs across an entire race. Two races, featuring different drivers with different strategies, have been plotted to illustrate the model's performance.

These plots display the model's prediction for each lap, with diamonds plotted at heights proportional to the model's confidence in a pit-stop being necessary on the following lap. A dotted horizontal line marks the 0.5 threshold: predictions above this value indicate a positive (make a pit stop), while predictions below it indicate a negative (do not make a pit stop). Vertical lines indicate the laps where drivers made actual pit-stop decisions. Additionally, a plot below tracks the drivers' positional changes throughout the race, offering a visualization of the success of the pit-stop strategies employed.



The plot above shows the model's pit-stop predictions for drivers Lewis Hamilton and Sergio Pérez during the 2024 Spanish Grand Prix. This allows us to see how the model responds to both a two-stop and a three-stop strategy during the same race. You can see how the model's outputs trend upwards as a pit stop approaches, which in real-time communicates to race engineers that a pit-stop window is beginning.

The model successfully adjusted to each strategy, accurately predicting 4 out of 5 stops. It only misses Perez's third pit stop, but it correctly trends upward around the time of the stop, indicating that the model was close to making the correct prediction or that Perez could have stayed out on track for longer.



The plot above shows the model's pit-stop predictions for drivers Carlos Sainz and Daniel Ricciardo during the 2024 Austrian Grand Prix. We again observe how the model responds to two different strategies. It accurately predicts 3 out of the 4 stops, only missing Ricciardo's first stop, which occurred very early in the race as part of an alternate strategy by his team. However, the model adjusts to this strategy and correctly predicts his final stop.

Data Summary

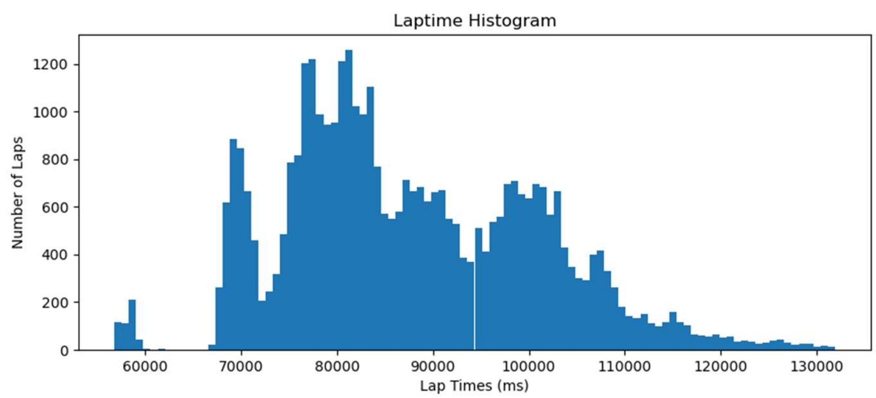
Data Source

The data used to train this neural network comes from Kaggle (Rao, 2024) and is sourced from the Ergast Motor Racing Developer API (Newell). This dataset includes several tables, with the most notable being a comprehensive lap-by-lap record of car positions and lap times from nearly every race. By joining this data with other tables—such as those containing race details, results, and pit stops—we obtained all the raw information necessary to train the network. However, before development could begin, significant preprocessing was required to extract and refine the useful information from the initial dataset.

Data Filtering

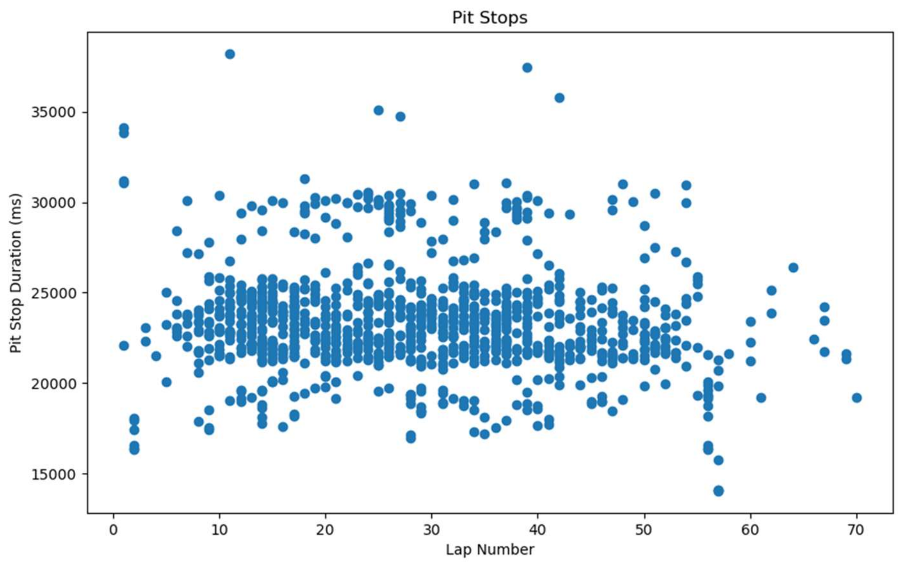
The data used to train the neural network consisted of lap-by-lap information from various drivers across entire races. If any lap contained bad data for any reason, the data for that driver's entire race was discarded. This approach ensured the network did not learn from strategies influenced by unusual or unpredictable circumstances. Data was classified as bad if it met any of the following conditions:

- Data older than 2014:
  This data predates the 'hybrid era' of Formula One, making it irrelevant to modern-day strategies.
- Driver didn't complete the race:
  If a driver fails to finish, it indicates something went wrong during the race, making the data less reliable for training.
- Finished outside the top 10:
  Retaining only data from drivers who finished in the top 10 ensures the model trains on strategies used by high-performing teams, which are more likely to be successful.
- Finished worse than their starting grid position:
  If a driver's finishing position was worse than their starting position, it suggests that the strategy employed was ineffective.
- Had a lap exceeding 2:12 minutes:
  Lap times longer than 2:12 minutes typically indicate unusual circumstances on most tracks and are therefore considered outliers.



A histogram of the remaining lap times is displayed above, revealing a fairly normal distribution, with peaks corresponding to tracks of varying lengths. The rightward tail in the distribution is due to slower laps at the start of the race and laps following pit stops, where cars are necessarily stationary for a period. However, an outlier group below 60 seconds, originating from the 2020 Sakhir Grand Prix, was identified and removed from the dataset.

- Pit stops longer than 32 seconds:
  These unusually long pit stops typically indicate that there was an issue with the car, likely requiring an altered strategy as a result.
- Pit stops before lap 5 or after lap 65:
  Pit stops made too early or too late in the race are generally not optimal strategies. Stopping at the start wastes the potential of fresh tires, and stopping near the end doesn't provide enough laps to fully benefit from new tires.



The scatter plot of pit stop duration versus the lap on which the pit stop was performed illustrates that early, late, and long pit stops are outliers among successful strategies.

Calculating Additional Metrics

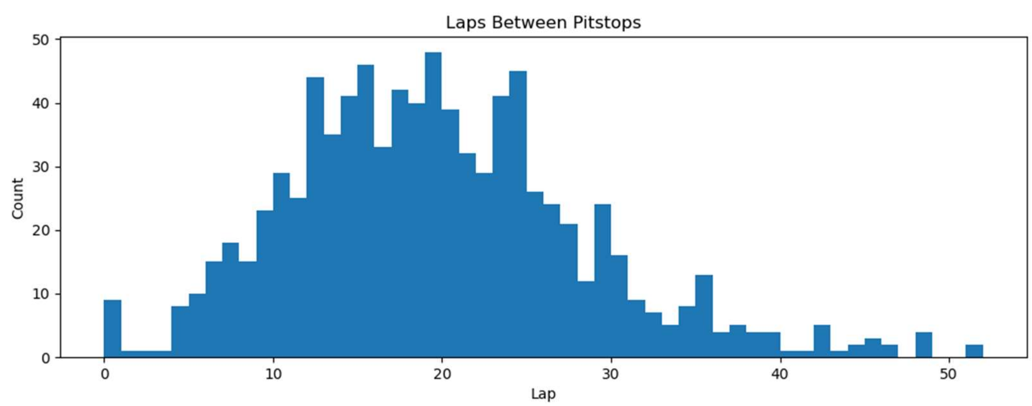
After filtering out the bad data, the next step in data processing involved calculating and generating additional columns of information needed for training the neural network. While not directly included in the source data, these additional data points could be derived from the existing information. Calculating these metrics was crucial, as they provided valuable insights that enhanced the model's predictive capability. The calculated information included the following:

- Laps since the last pit stop:
  This helps the model understand tire wear and the likelihood that a driver may need fresh tires.
- Race progress value between 0 and 1:
  This normalized value represents how much of the race is remaining, which is essential since different races have varying total lap counts.
- Time split from the previous lap:
  This helps the model assess if a car’s performance is improving or deteriorating.

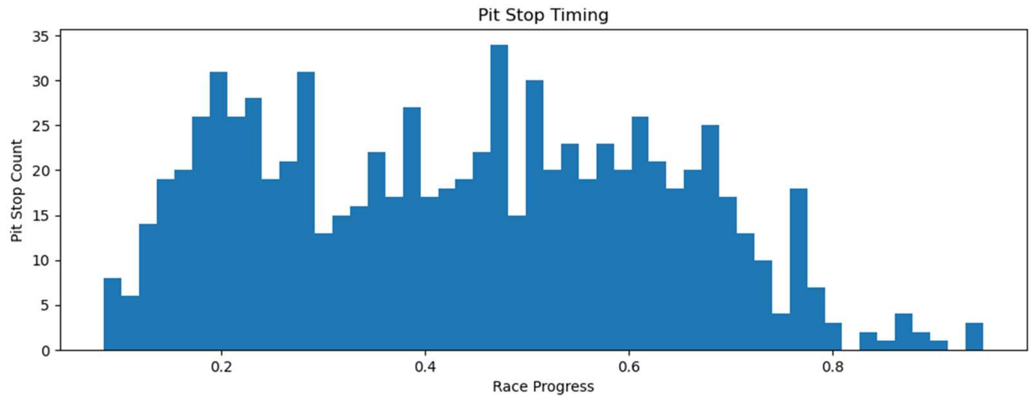
- Time gap to the car a position ahead:
  A smaller gap indicates the car might be in a good position to challenge for an overtake, which could influence the decision to pit.
- Is DRS available?
  Under the current rules, a driver can only use the Drag Reduction System (DRS) if they are within one second of the car ahead when passing specific points on the track. Whether or not a driver has access to DRS is an important factor to consider when deciding whether to make a pit stop.
- Time gap to the car a position behind:
  Similarly to the gap ahead, this gap indicates if the competitor behind can challenge our car.
- Did the car ahead make a pit stop?
  If the car ahead has pitted, it can influence strategy as it may present a new opportunity.
- Did the car behind make a pit stop?
  Similarly, if the car behind pits, it might be a good opportunity to pit as well, minimizing the risk of losing track position.

Strategy Analysis

With these additional metrics, two additional plots were created to get an overview of the average pit stop strategy.



This first histogram shows the average number of laps between pit stops, representing how long a car typically stays on the track before changing tires. The distribution appears to be roughly normal, with an average of around 18 laps. The wide range of values indicates that there isn't a single dominant pit stop strategy in terms of tire longevity, with different strategies being employed depending on various factors like track conditions and tire compounds. The outliers seen below 5 laps were likely the result of exceptional circumstances, such as early damage or safety car interruptions, and were subsequently removed.



This second histogram illustrates the points during a race at which pit stops were performed. Notably, there is a clear peak around 0.2 race progress, indicating that many teams tend to pit early in the race. However, apart from this peak, the distribution lacks a distinct structure which suggests that there are no universally dominant strategies employed across all races. This variability highlights the complexity of pit-stop decision-making.

The absence of a consistent pattern highlights the value of our neural network model. By learning from historical data, the model can identify critical moments when a pit stop was successful under similar circumstances, thereby providing race engineers with actionable insights to enhance our strategic decision-making in real-time.

Data for Training

After all the processing was completed, the data was saved to a file. Data from three races in the 2024 season were separated from the training set and later used as the testing set for the model. This ensured that the model was relevant and effective in a modern racing setting.

Post-Deployment

To perform maintenance on the application after deployment, it is expected that new data collected from races will be processed in the same way and added to the complete dataset. The code used to process this data will be provided along with the application. Any new data should then be used to retrain the model to ensure that it remains up to date with the latest strategies being used by other teams.

Machine Learning

Binary Classification

The race data available to us contained detailed information for every lap completed by each driver in every race. In addition, we had comprehensive information about every pit stop during these races. This data enabled the application of a supervised learning algorithm to train on various pit-stop strategies. Since the decision to make a pit stop during a lap is binary, the problem is best suited to a classification task rather than a regression, thus framing it as a binary classification problem.

Neural Network Architecture

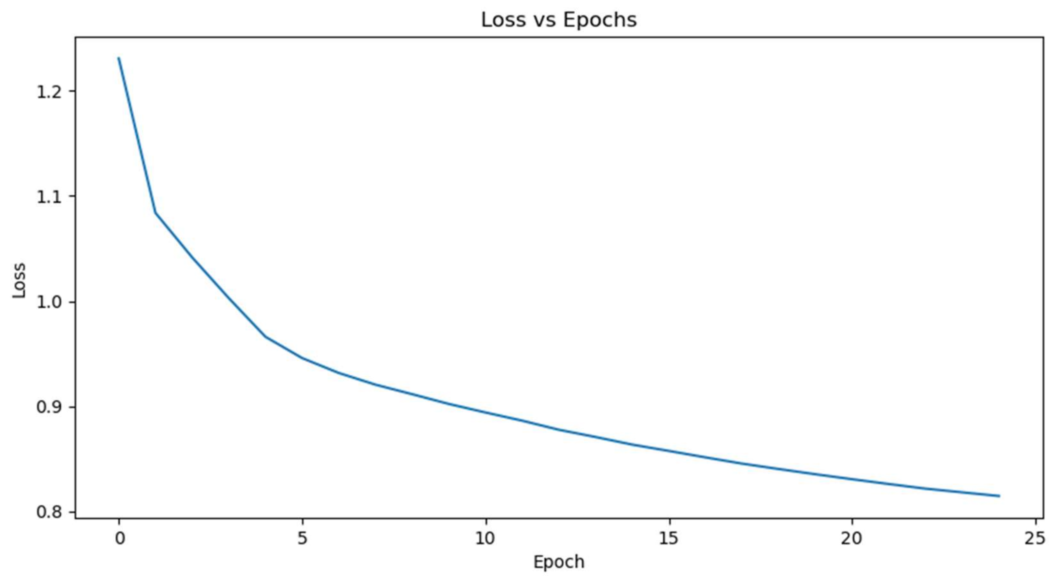
To solve this, we utilized a neural network for the machine learning algorithm—specifically, a hybrid architecture comprised of one long short-term memory (LSTM) layer and two fully connected layers with each layer containing 128 neurons. This hybrid combined the strengths of a recurrent neural network (RNN) and a feed-forward neural network (FFNN) to create a network optimally suited for this application.

The choice of this architecture came after experimenting with both a pure LSTM network and a pure FFNN on the same dataset. The FFNN, while having a low loss on the test set, suffered from overfitting and produced sporadic results. Its lack of memory resulted in values that jumped suddenly from lap to lap and did not trend in a way that would provide meaningful insights for race engineers. The results from the pure LSTM were much better in that regard. Its predictions steadily rose as pit stops approached and reset after a stop, offering a more realistic and actionable output.

Combining the output of the LSTM into a FFNN produced the most reliable and insightful results. This hybrid approach worked best for this application because it integrated both long-term memory and accurate pattern fitting—two essential features when evaluating pit-stop decisions. A car's performance over multiple laps needed to be analyzed to determine the best time for a pit stop, making long-term memory crucial for accurate and consistent predictions. The LSTM layer provided this memory, while the FFNN ensured precise pattern recognition.

Training

After establishing the hybrid model, the next step was to train the network effectively. Even with an ideal network structure, perfecting the training process is a crucial part of producing an accurate model. While plotting loss versus epoch during training might suggest that the model could still improve by continuing to train, doing so comes with the risk of overfitting. In practice, when training the model for many additional epochs, the result was a network that performed poorly on the test data due to overfitting.



This graph plots the loss for each epoch of training. It shows that the network was effectively learning over time.

Validation

The network's performance was validated using three key metrics for binary classification: precision, recall, and F1-score. Precision measures the fraction of positive predictions made by the model that were true positives. Recall measures the fraction of positive cases in the data set that were identified correctly. The F1-score is the harmonic mean of precision and recall, providing a balanced assessment of both metrics.

To evaluate the model's performance, it was tested on a dataset containing data from three races in the ongoing 2024 Formula 1 season. The model's outputs were classified as positive if the value exceeded 0.5 and negative otherwise. The results for each metric were as follows:

- Precision: 0.183
- Recall: 0.706
- F1-score: 0.290

Of these metrics, the recall was prioritized because it is more important for the model to correctly identify a pit stop window, even if slightly early than to miss it entirely by being overly precise. A high recall ensures that race engineers are alerted when a pit stop is approaching, even if the timing isn't perfect. Despite the low precision, the model's high recall (correctly identifying 70.6% of the test data pit stops) makes it a useful tool for race engineers.

Visualizations

Visualization can be found in this post-implementation report in the following locations:

- Solution Summary: Results (pg. 10-11)

  These visualizations show the outputs of the completed model on test data

- Data Summary: Data Filtering (pg. 12-13)

  These visualizations show the distribution of lap times and pit stops.

- Data Summary: Strategy Analysis (pg. 14-15)

  These visualizations show the distribution of laps between pit stops and the points during races where pit stops were made.

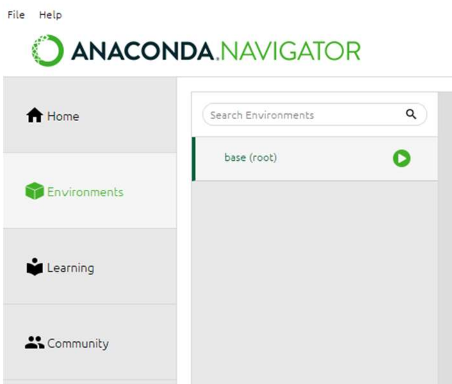
- Machine Learning: Training (pg. 17)

  This graph shows the progress of training of the neural network

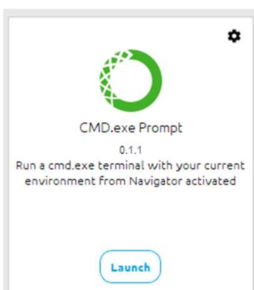
User Guide

Installation

1. Install the Python distribution, Anaconda, from www.anaconda.com
2. Open the Anaconda Navigator and ensure you are in the Environment you wish to use (base is the default)

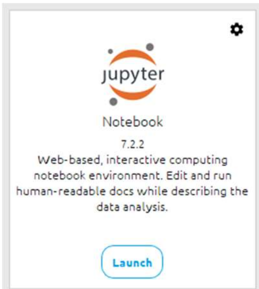


3. Go to 'Home' and Launch the CMD.exe Prompt



4. Make sure the necessary packages are installed by running the following commands in the console:
    a. conda install anaconda::numpy
    b. conda install anaconda::pandas
    c. conda install anaconda::scikit-learn
    d. conda install anaconda::ipython
    e. conda install anaconda::ipywidgets
    f. conda install pytorch::pytorch
    g. conda install conda-forge::matplotlib

5. Make sure everything is updated by running the following command:
    a. conda update --all
6. Close the console and return to Anaconda Navigator
7. Go to 'Home' and Launch Juypter Notebook

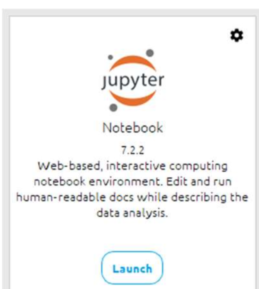


8. This should automatically open your default web browser and navigate to 'localhost:8888/tree'. If not, do so manually to access the Jupyter Notebook server.

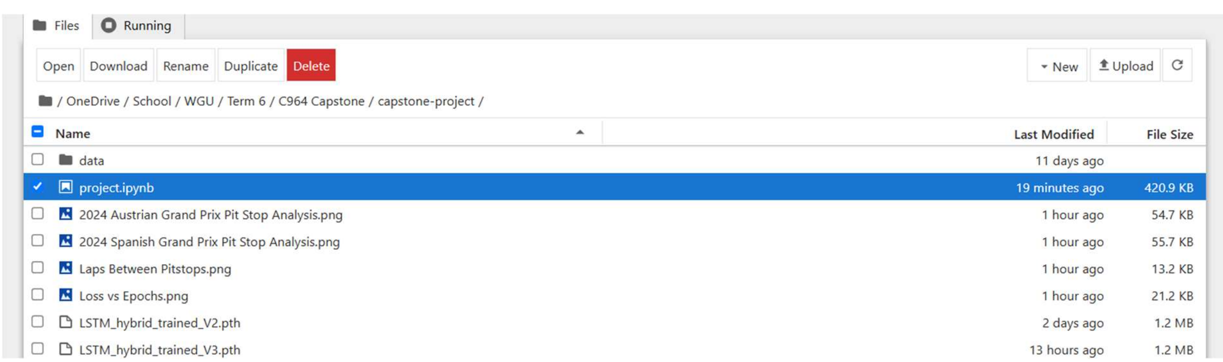
This confirms the installation was successful.

Running the Program

1. In Anaconda Navigator, go to 'Home' and Launch Juypter Notebook



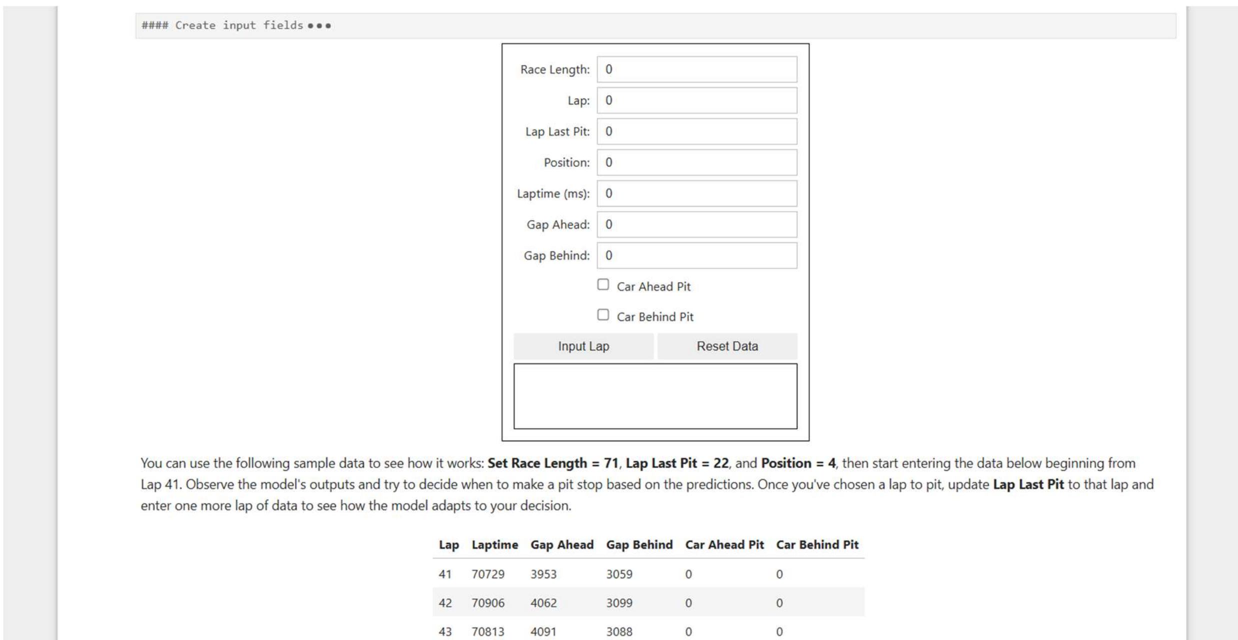
2. This should automatically open your default web browser and navigate to 'localhost:8888/tree'. If not, do so manually to access the Jupyter Notebook server.
3. From this file tree, navigate to the directory where you saved the project and open the file 'project.ipynb' *(Note: your file path and directory contents will be different)*



4. Once open, **DO NOT** run all cells in the project. The computation time to process all the cells will be long.

5. Scroll to the bottom of the project to the section titled 'User Interface' and run only the cells that are in that section. *(You can do this by clicking on the 'User Interface' cell and then pressing SHIFT+ENTERT for each cell until reaching the end of the project)*

6. Locate the input fields that were generated in the 'User Interface' section, and enter the desired data, lap by lap. *(Example data has been provided at the bottom of the project for testing)*



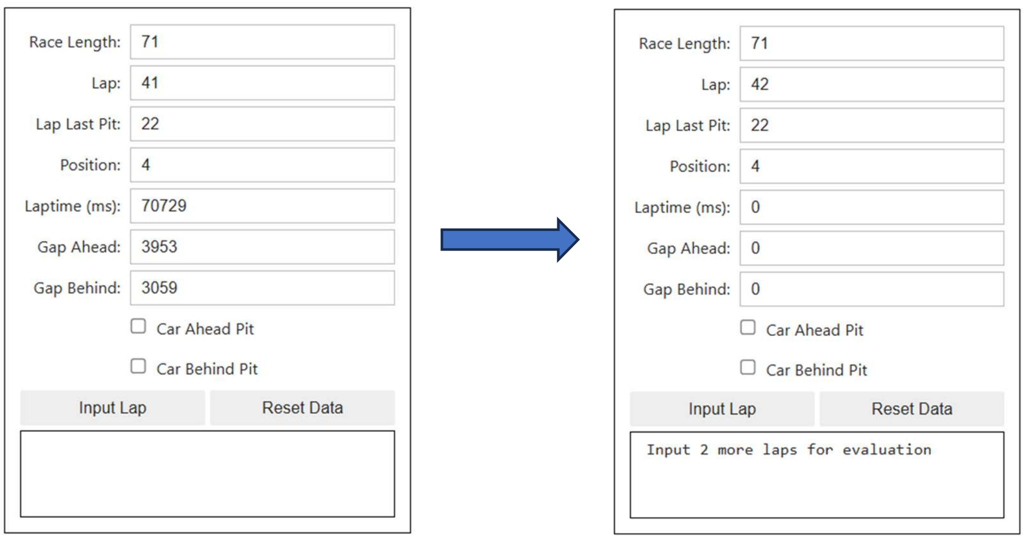
Input Fields

Metric	Description
Race Length	The total number of laps in a given race
Lap	The lap currently being evaluated
Lap Last Pit	The lap that a pit stop was last performed
Position	The current place of the driver in the field (1st, 2nd, 3rd...)
Laptime (ms)	The time taken to complete this lap, given in milliseconds
Gap Ahead	The time gap to the car a position ahead, given in milliseconds
Gap Behind	The time gap to the car a position behind, given in milliseconds
Car Ahead Pit	Did the car ahead make a pit stop?
Car Behind Pit	Did the car behind make a pit stop?

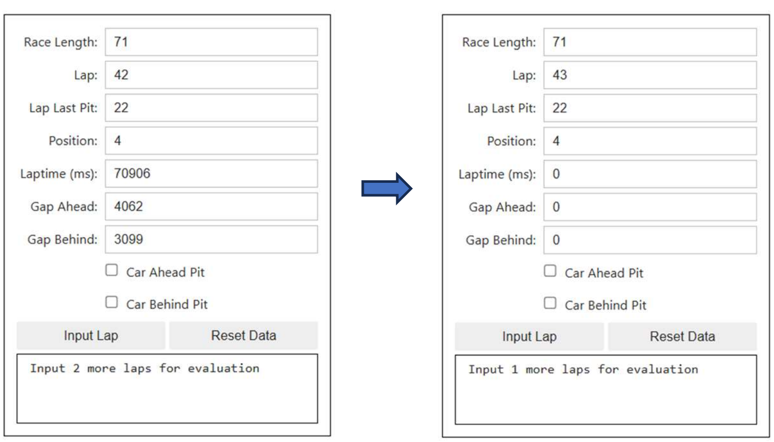
- For any race you wish to evaluate, whether during or after the event, it's important to enter the lap data in the correct sequence.
- Be sure not to change the Race Length without resetting the data first.
- You can begin evaluating from any lap, but you won't be able to go back to previous laps once you've started.
- The model needs **3 laps** of data before it can produce any output, so no prediction will appear until at least 3 laps have been input.
- Each time you press "Input Lap," the model will take the data from the input fields and automatically increment the Lap field for you.

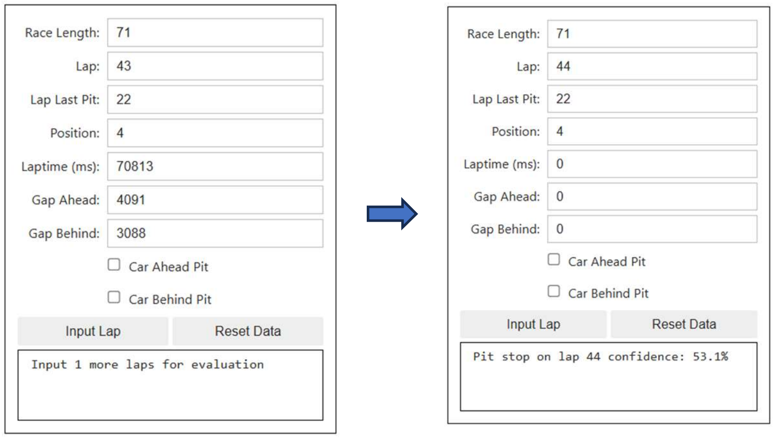
Example Usage

1. Input all the information from the first lap of your data and then press "Input Lap"

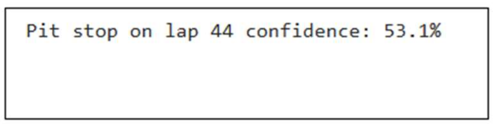


2. Continue to add lap data by inputting **Laptime**, **Gap Ahead**, and **Gap Behind** for each successive lap and changing **Position**, **Car Ahead Pit**, and **Car Behind Pit** when necessary.





3. After three laps have been input into the model, it will begin outputting predictions on the likelihood of a pit stop being made during the next lap.



4. When your car makes a pit stop, change the **Lap Last Pit** to the lap the pit stop occurred and continue to input data.

Note: a subset of example data has been provided at the bottom of the Jupyter Notebook for the user to test the application

REFERENCES

Rao, R. (2024). *Formula 1 World Championship (1950 - 2024).* Retrieved October 3, 2024, from https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020/data

Newell, C. (n.d.). Ergast Motor Racing Developer API. http://ergast.com/mrd