Predicting Formula 1 Race Outcomes: A Machine Learning Approach

Ali Jafri
Computer Science, NYUAD
aj3218@nyu.edu

Advised by: Djellel Difallah, Talal Rahwan

ABSTRACT

This project explores the application of machine learning to predict Formula 1 lap times and race outcomes, leveraging historical data spanning from 2014 to 2023. The primary objective is to develop a predictive model capable of accurately forecasting lap times for individual drivers throughout a race. By utilizing advanced machine learning techniques, particularly Long Short-Term Memory (LSTM) networks, this research aims to uncover the complex relationships between various factors influencing lap times, such as track characteristics, team performance, driver performance, and historical trends.

The study begins with a comparative analysis of simpler models like linear regression, decision trees, and random forests to establish baseline performances. LSTM models are then employed to handle sequential data and capture temporal dependencies in lap times. The results demonstrate the effectiveness of LSTMs in predicting relative driver performance and race outcomes, with refinements such as custom loss functions improving accuracy. Findings from preliminary tests revealed that minimizing laptime loss alone does not always translate to better race outcome predictions.

The evaluation includes testing on the 2023 Abu Dhabi Grand Prix and 2024 Bahrain and Abu Dhabi Grands Prix, showcasing significant improvements in prediction accuracy with later model iterations. Future work will explore transformer models for their ability to capture global dependencies and extend prediction scope to include pit stop strategies and safety car laps. This research not only advances predictive analytics in motorsport but also offers valuable insights into Formula 1 dynamics, enhancing engagement for teams, broadcasters, and spectators alike.

KEYWORDS

Formula 1, machine learning, motorsport, lap analysis, laptime prediction race prediction, LSTM, transformers, regression

Reference Format:
Ali Jafri. 2024. Predicting Formula 1 Race Outcomes: A Machine Learning Approach. In *NYUAD Capstone Seminar Reports, Spring 2024, Abu Dhabi, UAE.* 10 pages.

1 INTRODUCTION

Formula 1 racing stands at the pinnacle of motorsport, where drivers push the limits of human and machine performance on some of the world's most challenging circuits. Central to the competitive landscape of Formula 1 is the quest for optimal lap times, which directly influence race outcomes and hence affect championship standings. In this context, the prediction of lap times emerges as a critical endeavor, offering insights into driver performance, race strategies, and ultimately, the determination of race winners [1, 4].

The importance of accurately predicting Formula 1 lap times cannot be overstated. In the high-stakes environment of such elite motorsport, by forecasting lap times, teams gain a strategic edge in race planning, pit stop optimization, and tire management, all of which are crucial factors in securing victory. Moreover, for broadcasters and spectators, lap time predictions add an extra layer of excitement and engagement, enabling informed speculation and analysis throughout the race. Having a tool that can assist with making such educated guesses not only demystifies such a complex sport but can also deepen the sense of involvement for all audiences.

To address the challenge of lap time prediction in Formula 1, this research builds upon a foundation of data-driven analysis and machine learning methodologies. Drawing from extensive datasets spanning multiple seasons and race circuits, the project aims to uncover the complex relationships between various factors influencing lap times, including track

This report is submitted to NYUAD's capstone repository in fulfillment of NYUAD's Computer Science major graduation requirements.

جامعة نيويورك أبوظبي
NYU ABU DHABI

Capstone Seminar, Spring 2024, Abu Dhabi, UAE

characteristics, car performance, and driver behavior. Therefore the main research question of this project is 'How accurately can we predict lap times of a given race using machine learning methodologies?'.

However, the task of predicting Formula 1 lap times presents several formidable challenges. The dynamic nature of racing environments, the interplay of multiple variables, and the inherent uncertainty of such a competitive sport pose significant obstacles to accurate forecasting. Yet, it is precisely these challenges that make this work both hard and exciting. Success in this endeavor promises not only advancements in predictive analytics but also deeper insights into the intricate dynamics of Formula 1 racing, perhaps even pushing the boundaries of what is achievable in sports analytics and machine learning.

2 RELATED WORK

2.1 Machine learning analysis

An important utilization of machine learning techniques is to find the importance of certain variables for determining a particular outcome. That is exactly what one paper has done, by using tree-based models to find the relevance of variables such as starting position, constructor points, and driver points in terms of finishing top 3 in a given race [12]. Another study that was done had a similar approach, where instead of using tree-based models, it utilized LSTMs to find explanatory variables. The findings of this paper were essentially that the performance of a driver in the qualifying session is crucial to determining the end position of said driver in a race [13]. The work done by both of these papers will be useful for this project as it will allow the choosing of the correct variables for the dataset, and make sure that any key variables are not left out.

2.2 Predicting or Optimizing Strategy

Along with the analysis of what variables are important, another use of machine learning in this context is to find optimal solutions in order to maximize performance. In this case, machine learning could be used to determine optimal pit stop strategies. This is exactly what is proposed in a Master's thesis, whereby the author aims to automate the identification of tire strategies for Formula 1 races by treating the problem as a sequential decision-making process. What is interesting about this approach is that the paper designs a planning environment that replicates past Formula 1 races using a lap time simulator based on regression techniques applied to publicly available race data [10]. This is quite interesting as it allows one to see how making a certain decision for a certain driver can change the whole race's landscape. Another paper used a slightly different approach, which was to use a predictive model for determining the optimal timing of pit stops during Formula 1 races. The author utilized 3 different machine learning algorithms (Support Vector Machines (SVM), Random Forest, and Artificial Neural Networks) to determine which one would be best in not only predicting whether a driver makes a pitstop at a given lap of the race, but whether the pitstop was a 'Good Pitstop' [8]. Such an approach would definitely be of use when trying to create a predictive model that not only takes into account raw driver performance but also the pitstop strategies of different teams.

2.3 Predicting race outcomes

Perhaps one of the most important applications of machine learning in the realm of Formula 1 is the prediction of race outcomes. One paper does exactly that by trying to predict the winner of Formula 1 races using Python and Support Vector Machines. Their model was trained using historical data from F1 races, such as lap times, sector times, qualification times, and information about the drivers and teams. They set their experiment as a classification problem, where they would determine whether a driver would finish on the podium, finish in the points (top 10 on the grid) or would not finish in the points (placed 11th to 20th position on the grid) [11].

Another paper that tries to predict race outcomes uses generalized additive models (GAM) to represent the evolution of lap times in Formula 1 races. Their aim is to analyze Formula 1 team and driver performances during a specific race by modeling lap times as a function of relevant predictor variables, both numeric and categorical. The study focuses on the Formula 1 season of 2015 and utilizes freely available data to fit the model. Their results indicate that the model accurately describes race development in Grand Prix events without unpredictable occurrences such as safety car interventions or race suspensions [6]. Additionally, their model shows potential for specifying alternative race strategies, particularly concerning pit stop choices. The approach specified in the paper is especially valuable to the current project, as it predicts specific laptimes rather than general results (as was seen in the previous paper). However, their data pool does not consider the 8-9 additional years of racing that happened after 2015, meaning that a newer model has the potential to be even more accurate.

3 METHODOLOGY

3.1 Gathering the dataset

The first step in this project was to gather all the necessary data required to train the models. There are multiple APIs available that can be used to source this data. One of them being the Ergast Developer API. This API has the lap times for

all drivers for all races in a given season. While the API also has some other data available, like the number of pitstops, it doesn't provide the full details such as what tire compound was on before and after the stop. Another popular API is FastF1, which rather than providing a XML or JSON response (like how the Ergast API usually does) provides a Python library where all the data can be accessed from there. This API offers more than the Ergast API, as not only does it have lap times, but also has the tire strategies used by all drivers in a given race and the laps which were held under a yellow flag or safety car. However, the data for the FastF1 API is limited in that there is no tire compound data before 2018. For additional data, such as finding the number of wins of a driver up to a given point in the season, the StatsF1 website was used. Using all of these APIs and alternate methods will result in a raw dataset that includes data from 2014 all the way up to 2023.

The dataset comprised information from 203 races, 55 unique drivers, and a total of 214607 laps. Each row gives you information on a lap completed by a certain driver in a given race. The dataset can be split into three different categories: core features, race conditions, and driver and team flags. The core features consist of things like race and driver identifiers (the ID of the race, driver, constructor, etc.), performance metrics (grid position and qualifying lap times), and driver statistics (number of total wins, races completed, podiums, points this season, etc.). The race conditions consist of safety car indicators (whether a safety car was deployed during or before a lap) and pit stop information (whether the driver is going to pit this lap, the compound of the tire, and the age of the tire). The driver and team flags (in the format of isHAM to signify that this is Lewis Hamilton's lap, or isRBR to signify that this lap is of a driver who races for Red Bull) indicate which driver or team the lap corresponds to. The target variable is the time (in milliseconds) that the driver took to complete that lap. This is the primary outcome that the model aims to predict using the features mentioned above.

3.2 Comparative and Reproducibility study

In this section, we delve into the comparative analysis of various machine learning models and their applicability to predicting Formula 1 race outcomes. This analysis involves understanding the strengths and weaknesses of different models, reproducing results from existing projects, and discussing performance outcomes. The selection of models for this project is based on their potential to handle the complexities of Formula 1 data. The models considered include: linear regression, logistic regression, decision tree models, random forest models, LSTMs, and transformers.

For linear regression, it was seen that it was suitable for predicting continuous variables like lap times. While this is easy to interpret and fast to train, it assumes linear relationships which may not capture the intricate dynamics of racing data. Nevertheless, this could still be a potential candidate for predicting the lap times of the drivers for a race. Logistic regression was looked at for its potential with binary classification tasks like predicting whether a driver pits, whether there is a safety car, or (on a more macro-scale) whether a driver finishes on the podium [9]. A potential good use for this is also multi-class predictions, such as what tire (out of the three compounds available) a driver will put on when they come in for a pitstop. However, since the main focus of this project is to predict lap times, the maximum that classification models can do is act as a supplementary layer that predicts safety cars, pitstops, and tire compound usage. This predicted information can then be used to predict lap times.

Decision tree models are also a good potential option as they can do both regression and classification tasks. Moreover, they can capture complex relationships (where linear regression and logistic regression may not be able to). However, based on research, it seems that decision tree models are quite prone to overfitting, especially with noisy data [7]. This is where random forest models might be better, as they can still utilize decision tree models' ability to capture complex relationships, but can reduce overfitting by averaging over multiple trees. However, it would be harder to interpret the model and understand the importance of certain features of the dataset in predicting the desired outcomes.

Long Short-Term Memory Networks (also known as LSTMs) are another potential candidate for predicting Formula 1 lap times. These models are ideal for time-series prediction and capturing long-term dependencies in sequential data like lap times. Since lap times are sequential and depend on previous laps' conditions, LSTMs can be effective by learning the temporal relationships. Moreover, this model type has been used in other projects for very similar use cases (Jared Chan, 2021). Another potential model type is transformers, which use time-series prediction (like LSTMs) but are more powerful in capturing global dependencies over long sequences. However, for these model types (both LSTMs and transformers) the complexity of setting up the model and training the models is significantly higher compared to the previous models mentioned. Additionally, transformers specifically are known to work well with particularly long sequences [14] and for the use case that this project explores, we are limited to only 50 to 70 laps for a given race.

Now that we have our dataset and also an understanding of the models that can be used, we shall now try and reproduce the results of two existing projects with the dataset that was made for this project. Starting with Veronica Nigro's project,

we can see that her aim is to predict race outcomes like who wins the race. She used data from 1983 to 2018 (using the 2019 season as the testing data) and included things like race information, results, weather, driver and team standings, and qualifying times. She had originally looked into logistic and linear regressions, random forests, support vector machines, and neural networks from both a regression and classification perspective. Upon doing some testing and tuning she found that the neural networks classifier model was the best suited and gave the best results, correctly predicting the winner for 62% of the races in 2019. Using the same tuning parameters but with a slightly modified version of the dataset meant for this project, we were able to get a score of 90% for the 2019 season (training the model only on races from 2014-2018). If the tuning parameters were modified slightly (changing hidden layer size and alpha value) we got an accuracy of 86% for the 2022 season (using up to the 2021 season as training only). For 2023, (training upto 2022) we got a score of 95%. Although this seems quite high, it must be noted that the 2023 season would have been very predictable given that Max Verstappen had won almost all of the races. Nevertheless, we have seen from our results of 2019 that the model that was trained using the dataset for this project performed significantly better (even with the same training parameters as the final model by Veronica Nigro). Hence, we can say that even if the approaches to training are similar, in the end, the dataset plays a crucial role in determining the model's performance.

The second project we look at is by Jared Chan and uses LSTMs to predict lap times, positions, and pit stop strategies for up to 20 drivers throughout a race. He uses data from 2001 to 2019, using 2020 as the testing data, and has an interesting structure for his dataset, whereby each record in his data for a given race contains the lap information for all 20 drivers. This information for each driver includes the driver's ID, the driver's standing, the constructor's standing, the driver's status, the driver's race position, whether the driver is pitting, and the lap time (the last three being the target variables that are going to be predicted by the model). To reproduce these results with the dataset meant for this project meant that a lot of modifications would be needed not only the data we had, but also how it would be used by Chan's code for training and evaluating. When evaluating his model, it received an average loss score of 34.34, while the model trained using the restructured version of our dataset got a loss score of 39.35. While the units are arbitrary as the data was scaled and averaged between the three target variables, we can see that the model that used our dataset doesn't perform as well as the original project's model. A potential reason for this is that the setup of his model and helper functions must have been optimized for the structure of his dataset. Our dataset has a bit more variables per driver (for example the driver statistics which contain the number of wins a driver has), and we were also not focusing on the status variable of the laps, which might have played a negative role. Moreover, when seeing the actual predictions from the model using our dataset, there were some outlandish predictions. For example, we were getting results that Latifi (a driver who is known to barely ever finish in the top 10 and is usually part of the bottom half of the grid [5]) was getting podiums. So results like this might not seem outlandish to the model, but this is a huge mistake to anyone with any knowledge of F1 context.

Based on the findings from the comparative and reproducibility study, it is evident that focusing on LSTMs (Long Short-Term Memory networks) and using relatively simpler models (linear regression, decision trees, or random forests) as baselines is the best route for this project. The primary reason for focusing on LSTMs lies in their ability to handle sequential data effectively, which is a key characteristic of Formula 1 lap times. Unlike simpler models like linear regression, which struggle with capturing complex temporal dependencies, LSTMs excel in modeling time-series data by learning relationships across laps. The reasonable success of Jared Chan's project further reinforces the potential of LSTMs for this use case, as his model demonstrated adequate performance in predicting lap times, positions, and pit stop strategies by leveraging temporal patterns in race data. While reproducing his results using our dataset revealed some challenges, such as differences in dataset structure and outlier predictions, these issues highlight the importance of tailoring the model to the specific characteristics of our dataset. We have also seen from Veronica Nigro's project that how much of an increase in performance can happen just by changing the dataset. Although LSTMs have been explored in similar contexts before, their application using this enriched dataset provides a novel opportunity to fully realize their potential in capturing the complex dynamics of Formula 1 racing. By leveraging the unique features of our dataset—such as detailed driver statistics and tire strategies—this project has the opportunity to push the boundaries of what LSTMs can achieve in Formula 1 lap time prediction.

3.3 Setting up the baseline model

To choose the baseline model we first need to find out which model (out of linear regression, decision trees, and random forests) performs the best using the default parameters set by the Sci-kit Learn package. The dataset used was split into training and test data, with 2014 to 2022 being the training data and the 2023 season being the test data. The data was also scaled using the StandardScaler function in the Sci-kit Learn package in order to standardize the input data such that the data points have a balanced scale. The inputs were the same structure as the original dataset (where each row

gives you information on a lap completed by a certain driver in a given race) and hence used all the core features, race conditions, and driver and team flags. The output was the lap time in milliseconds for that particular lap. Each of the models were trained and then tested on the 2023 season data, getting the necessary scores in the form of mean absolute error, mean squared error, root mean squared error, and the r-squared score.

3.4 Setting up the LSTM model

For the initial setup, we had used the original structure of the dataset without any modifications, and scaled the features and the target using the StandardScaler by Sci-kit Learn. Once scaled, the data was arranged in sequences such that each sequence item was a list of the laps completed by a certain driver in a certain race. The targets (the lap times) were also arranged ina similar array, where each item was the list of the lap times for a certain driver in a certain race. Then these were put on PyTorch tensors and padded to ensure consistent lengths. This step was important because not all the races are of the same length as some can be 44 laps while others can be 78. Then a dataset class was initialized with the sequence and target tensors and would return a particular driver's sequence of laps, lap times, and the length of their laps sequence (essentially the number of laps completed in a given race). The length is important for packing the padded sequence during training so that the model is not trained on the padding values. PyTorch's DataLoader was used to load batches of sequences during training. For the loss function, PyTorch's mean squared error function (MSELoss) was used. The models at this stage were trained with a hidden layer size of 128, 1 LSTM layer, no dropout, a learning rate of 0.001, a DataLoader batch size of 32, and only 10 epochs.

After a few training iterations and experimenting with different parameters, several improvements were made to how the training was done. For example, although 10 epochs were sufficient for Jared Chan's project, it was later realized that 10 epochs were too low for this project. This could be due to the fact that we are dealing with a smaller range of data compared to the other project. Thus, it was increased to 50 epochs with an early stopping mechanism added with a patience value of 10. This meant that if after 10 epochs the loss value hasn't gone down, then the model may have reached its plateau in terms of performance. Spending additional epochs for training might cause overfitting and be a waste of computational resources. Additionally, dropout was introduced to ensure that no overfitting occurred and the model can generalize better. Moreover, more LSTM layers were added to increase the trained model's capability to represent complex functions and relationships in the data.

Another important modification was to employ a manual scaler instead of the StandardScaler. This had multiple benefits, one of which was that it made the loss output interpretable. When using the StandardScaler we would get a training loss value of (for the sake of example) 0.26 and a test loss (the loss seen during testing the model on the 2023 season data) of 0.07. While this may be useful for a relative context where we see how the loss goes up or down based on changing parameters, it doesn't really give an idea of how many seconds off the mark are the predicted lap times compared to the actual lap times. While with the custom scaler, we know exactly the units of the loss output giving a good idea of how well the model performs in more absolute terms. Another benefit of using a custom scaler is that the StandardScaler calculates scaling based on mean and standard deviation, which can be influenced by outliers in the dataset. The custom scaler avoids this issue by using fixed scaling factors instead of relying on the statistical properties of the data.

Additionally, the loss function was modified from simply using PyTorch's mean squared error function to a custom loss function that combines lap time loss, position loss, and historical loss. The lap time loss is essentially the same as the original loss, where we get the mean squared error between the actual and predicted lap times. Position loss calculates the relative positions of drivers using the predicted lap time sequences in a batch, and compares them with the actual positions (calculated with the actual lap time sequences). This was introduced because it was seen that the test loss (which used only the mean squared error for the lap time loss) was around 6400 seconds, giving a root mean square error of about 80 seconds. While this isn't the mean absolute error and is probably significantly inflated in comparison due to the squaring of outliers, we have a general idea of the magnitude of the loss. Keeping in mind that the difference between drivers per lap is usually less than 4 seconds [2], having a root mean square error of about 80 seconds means that the lap time predictions may not be able to accurately reflect the relative performance of the drivers within a race, hence highlighting the importance of position loss. The last component of the combined loss function was the historical loss, which penalizes predictions deviating from historical performance expectations. The historical performance expectations are derived from the driver's season points, total wins, and total podiums. This was introduced because during preliminary testing it was often seen that poor-performing drivers were somehow being predicted to reach podiums. For instance, it was seen that Logan Sargeant, a driver who is part of a team that is struggling and is known to basically always finish outside of the top 10 [3], was somehow being predicted as a winner. In contrast, Max Verstappen (a four-time world champion) was being predicted to finish outside the points.

Having a weighted average between lap time loss, position loss, and historical loss (such that the weight distribution of the individual losses can be a tuning parameter) ensures that along with lap times, relative and historical performance is also taken into account.

3.5 Evaluation Methods

To get a deeper understanding of how the model performs it would be useful to see its predictions for certain races. For this, it would be ideal to choose a circuit that doesn't have as many incidents on track and is relatively easier to predict. These track incidents can range from weather changes (meaning that drivers and teams have to adapt their strategy for the wet climate mid-race), tire punctures, collisions, or even DNFs. Based on previous analysis[15], the Abu Dhabi Grand Prix is the race with the lowest average DNFs per season. Hence it makes sense to view the predictions of the 2023 Abu Dhabi Grand Prix to visualize the models' performance. To analyze the performance of the models for 2024, we will be looking at the Bahrain Grand Prix as well as it is also one of the circuits with the lowest average DNFs per season.

When viewing the predictions we shall see how the predicted final race position compares to the actual race positions. The race position is not a separate prediction output but rather inferred from the lap times themselves. The sum of the lap times for an individual driver would give the total race time, which can then be sorted in ascending order for all drivers to find out who finished the race in what order. Hence the total race time is directly linked with the final race positions. In order to compare the predicted positions with the actual positions, we can use four different metrics. The first three metrics are how many of the top 10, top 5, and top 3 in the predicted and actual positions are the same (irrespective of order). The fourth metric is whether the predicted and actual positions have the same winner. If we were to take order into account, this would be quite an unrealistic expectation for the model to predict the exact order for the top 10 correctly. Therefore we have four degrees of accuracy, where the first three are quite flexible as they don't consider order.

4 RESULTS

Now that we have seen how the models were set up, we can start by seeing the results from the linear regression, decision tree, and random forest models.

What we can see from Table 1 is that the random forest model performed the best by all metrics and hence will be used as a baseline to compare the LSTM models. Interestingly, we see that the mean absolute error is about 9.4 seconds (the original units in the table are set to milliseconds). So while this may not seem like a lot, if we reiterate the point of drivers only being about 3 seconds apart from each other in a race, the presence of this magnitude of an error might mean that the model fails to capture the relative performance of the drivers (especially since these models were trained with default parameters and without any custom loss functions).

Table 2.1 shows the predicted positions from various models compared to the actual positions of drivers for the 2023 Abu Dhabi Grand Prix. The models include the baseline random forest model and three LSTM-based models. LSTM model v1 signifies that the models were trained on data from 2014 to 2022 and tested on 2023 data. The v1.1 and v1.2 signify that they were trained using different configurations (different hidden layer sizes, different weights for the combined loss function, etc.). The results are then evaluated based on key metrics such as whether the correct winner was predicted, the number of podium matches, top 5 matches, and top 10 matches. What we can see is that firstly, none of the models correctly predicted Verstappen as the winner, highlighting a limitation in capturing dominant driver performances accurately. Additionally, some of the outlier predictions for each model have been highlighted. We can see that the baseline had quite a few outliers, as it predicted Sargeant winning the race and Zhou performing well, while Verstappen and Perez finish towards the bottom of the pack. The LSTM models v1.1 and v1.2 also had an outlier of predicting Sargeant too high, with v1.2 also predicting him winning the race. The LSTM model with the lowest test loss (which was just the lap time loss) also predicted Sargeant winning the race while Piastri finishing towards the bottom. If we look at Table 2.2, we can see that the LSTM models v1.1 and v1.2 performed significantly better than the baseline model. We can also see that the model with the lowest test loss (which may imply that this might have the most accurate predictions) performs worse than models v1.1 and v1.2. This means that a lower lap time loss does not always translate to better race outcome predictions, even though the final race outcomes are measured using lap times (by summing all the lap times together to get total race time). This goes back to the earlier point about how just focusing on lap time loss might overlook the drivers' relative performance within a race.

After conducting the tests for the 2023 season, the LSTM models were then trained from 2014 to 2023, and then tested on the 2024 season. The same analysis was then done on the 2024 Bahrain Grand Prix, which can be seen in Tables 3.1 and 3.2. Immediately, we can see that the predictions are significantly better and are quite accurate. Especially for models v2.1 and v2.2, the predictions for the top 5 positions not only match but also follow the exact order as the actual positions. We also again see that the model with the lowest test loss or lap time loss was not the one that performed the best. For reference, the 'v2' signifies that the models were trained on data from 2014 to 2023.

Model Type	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R-squared Score
Linear Regression	12695.6254	5776411599.7532	76002.7078	0.0912
Decision Tree	10099.0927	5880952590.9297	76687.3692	0.0747
Random Forest	9437.2788	5066568167.7366	71179.8298	0.2028

Table 1: Performance metrics of different models for lap time prediction in the 2023 season

Position	Actual Results	Predicted Results (baseline)	Predicted Results (LSTM model v1.1)	Predicted Results (LSTM model v1.2)	Predicted Results (LSTM model v1 with lowest test loss for 2023)
1	Verstappen	Sargeant	Leclerc	Sargeant	Sargeant
2	Leclerc	Leclerc	Verstappen	Verstappen	Alonso
3	Russell	Piastri	Norris	Leclerc	Hamilton
4	Perez	Alonso	Piastri	Perez	Perez
5	Norris	Zhou	Perez	Russell	Russell
6	Piastri	Hulkenberg	Alonso	Norris	Verstappen
7	Alonso	Ocon	Russell	Alonso	Leclerc
8	Tsunoda	Gasly	Hamilton	Tsunoda	Ocon
9	Hamilton	Hamilton	Sargeant	Hamilton	Norris
10	Stroll	Norris	Tsunoda	Piastri	Ricciardo
11	Ricciardo	Russell	Ocon	Stroll	Stroll
12	Ocon	Ricciardo	Albon	Zhou	Tsunoda
13	Gasly	Tsunoda	Stroll	Albon	Hulkenberg
14	Albon	Stroll	Zhou	Ocon	Gasly
15	Hulkenberg	Verstappen	Hulkenberg	Hulkenberg	Piastri
16	Sargeant	Albon	Gasly	Ricciardo	Zhou
17	Zhou	Perez	Ricciardo	Gasly	Albon

Table 2.1: Comparison of actual results and predicted results for the 2023 Abu Dhabi Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
Baseline Model (random forest)	False	1	1	5
LSTM Model v1.1	False	2	4	9
LSTM Model v1.2	False	2	4	9
LSTM Model (v1 with lowest test loss for 2023)	False	0	2	7

Table 2.2: Performance comparison of different models for the 2023 Abu Dhabi Grand Prix.

For the results of 2024 Abu Dhabi Grand Prix, when we look at Tables 4.1 and 4.2 we can see that the results don't seem to be as accurate as the predictions for the 2024 Bahrain Grand Prix. One possible explanation is that there were significant incidents that happened at the start of the 2024 Abu Dhabi Grand Prix. For instance, Piastri and Verstappen were involved in a contact that caused them to spin out of control (but still managed to continue racing). Perez also spun due to contact but DNF'ed. Then later Piastri was involved in a minor contact with Colapinto. Due to these incidents, a lot of penalties were also handed out, causing the lap and race times of the affected drivers to further increase [3]. With this in mind, we can see that the models performed fairly

Position	Actual Results	Predicted Results (LSTM Model v2.1)	Predicted Results (LSTM Model v2.2)	Predicted Results (LSTM model with lowest test loss for 2024)
1	Verstappen	Verstappen	Verstappen	Verstappen
2	Perez	Perez	Perez	Alonso
3	Sainz	Sainz	Sainz	Stroll
4	Leclerc	Leclerc	Leclerc	Russell
5	Russell	Russell	Russell	Norris
6	Norris	Stroll	Alonso	Leclerc
7	Hamilton	Alonso	Stroll	Perez
8	Piastri	Norris	Norris	Sainz
9	Alonso	Piastri	Hamilton	Piastri
10	Stroll	Hamilton	Piastri	Hamilton

Table 3.1: Comparison of actual results and predicted results for the 2024 Bahrain Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
LSTM Model v2.1	True	3	5	10
LSTM Model v2.2	True	3	5	10
LSTM Model (with lowest test loss for 2024)	True	1	2	10

Table 3.2: Performance comparison of different models for the 2024 Bahrain Grand Prix.

Position	Actual Positions	Predicted Positions (LSTM Model v2.3)	Predicted Positions (LSTM Model v2.4)
1	Norris	Norris	Hamilton
2	Sainz	Sainz	Norris
3	Leclerc	Leclerc	Russell
4	Hamilton	Verstappen	Sainz
5	Russell	Hamilton	Leclerc
6	Verstappen	Piastri	Verstappen
7	Gasly	Russell	Piastri
8	Hulkenberg	Gasly	Gasly
9	Alonso	Alonso	Alonso
10	Piastri	Perez	Perez

Table 4.1: Comparison of actual results and predicted results for the 2024 Abu Dhabi Grand Prix using different models.

Model	Correct Winner	Podium Matches	Top 5 Matches	Top 10 Matches
LSTM Model v2.3	True	3	4	9
LSTM Model v2.4	False	1	5	9

Table 4.2: Performance comparison of different models for the 2024 Abu Dhabi Grand Prix.

well, with v2.3 still managing to predict the winner and the podium in the exact same order as the actual positions.

5 CONCLUSION

This project demonstrates the potential of machine learning in predicting Formula 1 lap times and race outcomes, offering valuable insights into the intricate dynamics of motorsport. By leveraging Formula 1 data from 2014 to 2023 and employing advanced machine learning techniques, specifically LSTMs, this research has successfully highlighted both the challenges and opportunities in applying predictive analytics to Formula 1 racing. The study began with a comparative analysis of various machine learning models which allowed us to later establish baseline performances with simpler models such as linear regression, decision trees, and random forests. These models provided a starting point for understanding the complexities of the dataset and served as benchmarks for evaluating more sophisticated approaches. The LSTM models, designed specifically to handle sequential data, demonstrated superior performance in capturing temporal dependencies and relative driver performance within races. However, the results also revealed that minimizing test loss alone does not always translate to better race outcome predictions. This insight underscores the importance of incorporating additional evaluation metrics, such as position loss and historical loss, to align model objectives with real-world racing dynamics. The results from testing on the Abu Dhabi Grand Prix and Bahrain Grand Prix further validated the effectiveness of the LSTM models. While earlier versions of the LSTM models struggled with outlier predictions and failed to capture dominant performances accurately, refinements in model architecture and training methodology—such as adding dropout layers, increasing epochs, and introducing custom loss functions—led to significant improvements. The later iterations (v2.1, v2.2, v2.3, and v2.4) demonstrated remarkable accuracy in predicting top positions and race outcomes for the 2024 Bahrain Grand Prix and 2024 Abu Dhabi Grand Prix, showcasing the potential of these models when given the right data. For future work, further research can include looking into the same subject scope through the primary lens of transformers instead of LSTMs. This may help by leveraging transformer models' ability to capture global dependencies for improved lap time predictions. Then, one can see whether the transformer models outperform the LSTM models we have seen so far. Moreover, future research can also try to increase the prediction scope. Right now, only lap times are being predicted, but for a model that can 'truly' predict lap times for a given race, one must also be able to predict pitstop strategy (pitting lap and tire compound) as well as safety car presence in a lap. Predicting these would result in a more complete predictive model that can predict the outcomes of a race before it has happened.

REFERENCES

[1] FORMULA 1. 2023. Beginner's Guide to F1. https://www.youtube.com/watch?v=Q-jjZMMxbZs

[2] Formula 1. 2024. F1 - The Official Home of Formula 1® Racing. https://www.formula1.com/en/results/2024/races/1252/abu-dhabi/fastest-laps

[3] Formula 1. 2024. Relive the action from the season finale in Abu Dhabi. https://www.formula1.com/en/latest/article/highlights-relive-the-action-from-the-season-finale-in-abu-dhabi-as-norris.6hiM6GURCnP0FRiaMfCVk3

[4] Chain Bear. 2017. Basics of F1 Race Strategy. https://www.youtube.com/watch?v=wqf-dJyU_WA

[5] Adam Cooper. 2022. Why Latifi is struggling with the F1 2022 Williams. https://www.autosport.com/f1/news/why-latifi-is-struggling-with-the-f1-2022-williams/10247263/

[6] Carla De Francesco, Luigi De Giovanni, Marco Ferrante, Giovanni Fonseca, Francesco Lisi, and Silvia Pontarollo. 2017. *Proceedings of MathSport International 2017 Conference*. Padova University Press. 87–96 pages. https://www.padovauniversitypress.it/en/publications/9788869380587

[7] Data Headhunters. 2024. Decision Trees vs Random Forests: Comparing Predictive Power. https://dataheadhunters.com/academy/decision-trees-vs-random-forests-comparing-predictive-power/

[8] García Loreto and Tejada. 2023. *Applying Machine Learning to Forecast Formula 1 Race Outcomes*. https://aaltodoc.aalto.fi/server/api/core/bitstreams/70d5a580-c282-4278-8462-94d061471546/content

[9] Veronica Nigro. 2020. Formula 1 Race Predictor. https://towardsdatascience.com/formula-1-race-predictor-5d4bfae887da

[10] D Politecnico and Milano. 2020. *Open Loop Planning for Formula 1 Race Strategy identification*. https://www.politesi.polimi.it/bitstream/10589/175624/3/2021_04_Piccinotti.pdf

[11] Priya Shelke, Anurag Pande, Srujan Kale, Yash Paralikar, and Atul Kulkarni. 2023. F1 Race Winner Predictor. In *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. 1–4. https://doi.org/10.1109/ICCUBEA58933.2023.10392224

[12] Léon Sobrie. 2020. *SIFTING THROUGH THE NOISE IN FORMULA ONE: PREDICTIVE PERFORMANCE OF TREE-BASED MODELS*. https://libstore.ugent.be/fulltxt/RUG01/002/837/806/RUG01-002837806_2020_0001_AC.pdf

[13] William Villegas-Ch, Joselin García-Ortiz, and Angel Jaramillo-Alcazar. 2023. An Approach Based on Recurrent Neural Networks and Interactive Visualization to Improve Explainability in AI Systems. *Big Data and Cognitive Computing* 7, 3 (2023). https://doi.org/10.3390/bdcc7030136

[14] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2023. *Transformers in Time Series: A Survey*. https://www.ijcai.org/proceedings/2023/0759.pdf

[15] Intan Dea Yutami. 2022. Formula 1 Races Analysis. https://intandeay.medium.com/f1-f11dcd91d025

A APPENDIX: ADDITIONAL FIGURES

	raceId	circuitId	driverId	constructorId	grid	year	round	lap	milliseconds	q1milli	q2milli	q3milli	Driver_Seasc	driverwins	YOB	Races_befor	Races_won	Podiums	isSafetyCar	isSafetyCarPrev	i
0	900	1	1	131	1	2014	1	1	106128	91.699	102.89	104.231	0	0	1985	129	22	53	0	0	
1	900	1	1	131	1	2014	1	2	100287	91.699	102.89	104.231	0	0	1985	129	22	53	0	0	
2	900	1	3	131	3	2014	1	1	102038	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
3	900	1	3	131	3	2014	1	2	97687	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
4	900	1	3	131	3	2014	1	3	95765	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
5	900	1	3	131	3	2014	1	4	94939	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
6	900	1	3	131	3	2014	1	5	95438	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
7	900	1	3	131	3	2014	1	6	94977	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
8	900	1	3	131	3	2014	1	7	95417	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
9	900	1	3	131	3	2014	1	8	94550	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
10	900	1	3	131	3	2014	1	9	94217	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
11	900	1	3	131	3	2014	1	10	94364	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
12	900	1	3	131	3	2014	1	11	95185	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
13	900	1	3	131	3	2014	1	12	134269	92.564	102.264	104.595	0	1	1985	132	3	15	1	0	
14	900	1	3	131	3	2014	1	13	149953	92.564	102.264	104.595	0	1	1985	132	3	15	1	1	
15	900	1	3	131	3	2014	1	14	141208	92.564	102.264	104.595	0	1	1985	132	3	15	1	1	
16	900	1	3	131	3	2014	1	15	134486	92.564	102.264	104.595	0	1	1985	132	3	15	1	1	
17	900	1	3	131	3	2014	1	16	93976	92.564	102.264	104.595	0	1	1985	132	3	15	1	1	
18	900	1	3	131	3	2014	1	17	94040	92.564	102.264	104.595	0	1	1985	132	3	15	0	1	
19	900	1	3	131	3	2014	1	18	93195	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
20	900	1	3	131	3	2014	1	19	92478	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
21	900	1	3	131	3	2014	1	20	93331	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
22	900	1	3	131	3	2014	1	21	92839	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
23	900	1	3	131	3	2014	1	22	93144	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
24	900	1	3	131	3	2014	1	23	93213	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
25	900	1	3	131	3	2014	1	24	93936	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
26	900	1	3	131	3	2014	1	25	93941	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
27	900	1	3	131	3	2014	1	26	94539	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
28	900	1	3	131	3	2014	1	27	95588	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
29	900	1	3	131	3	2014	1	28	95416	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
30	900	1	3	131	3	2014	1	29	94069	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
31	900	1	3	131	3	2014	1	30	94946	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
32	900	1	3	131	3	2014	1	31	94272	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
33	900	1	3	131	3	2014	1	32	95305	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
34	900	1	3	131	3	2014	1	33	94088	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	
35	900	1	3	131	3	2014	1	34	93979	92.564	102.264	104.595	0	1	1985	132	3	15	0	0	

Driver_Season_Points	driverwins	YOB	Races_before	Races_won	Podiums	isSafetyCar	isSafetyCarPrev	isPitting	tyre_age	tyre_compou	isVET	isZHO	isVER	isTSU	isSTR	isMSC	isSAR	isRIC	i
0	0	1985	129	22	53	0	0	0	1	-1	0	0	0	0	0	0	0	0	
0	0	1985	129	22	53	0	0	0	2	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	1	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	2	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	3	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	4	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	5	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	6	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	7	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	8	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	9	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	10	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	11	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	1	0	1	12	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	1	1	0	1	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	1	1	0	2	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	1	1	0	3	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	1	1	0	4	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	1	0	5	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	6	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	7	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	8	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	9	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	10	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	11	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	12	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	13	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	14	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	15	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	16	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	17	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	18	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	19	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	20	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	21	-1	0	0	0	0	0	0	0	0	
0	1	1985	132	3	15	0	0	0	22	-1	0	0	0	0	0	0	0	0	

Figure 1: Snapshot of the Dataset

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
hamilton	10.392419	10.154527	9.753321	9.527362	9.405726	9.338777	9.30169	9.28113	9.269845	9.263825	9.260807	9.259494	9.259137	9.259295	10.223306	10.859659	9.90342
alonso	9.910796	10.061363	9.67752	9.4635725	9.349444	9.285577	9.2492285	9.228513	9.216766	9.210158	9.206464	10.00323	10.748145	9.898217	9.581088	9.409502	9.311471
hulkenberg	10.264563	10.27868	9.830133	9.598249	9.475173	9.405444	9.364628	9.340119	9.324916	9.315064	9.308332	9.303477	10.037667	11.020971	10.056638	9.713204	9.5258465
perez	9.900034	9.955861	9.602288	9.411341	9.30403	9.2399025	9.200524	9.175637	9.159243	9.147849	9.139434	9.132824	9.127332	9.122543	9.118203	9.1141405	9.925471
ricciardo	10.393998	10.221999	9.80014	9.582739	9.466246	9.400266	10.134547	10.914937	10.06331	9.7332325	9.552446	9.448869	9.3895235	9.355864	9.337211	9.327393	9.322827
bottas	10.325883	10.094572	9.746552	9.561308	9.46222	9.4058485	9.372709	9.35271	9.340224	9.332044	9.32632	9.321979	9.318395	9.315195	9.31216	9.30915	9.306078
kevin_magnu	10.553107	10.263282	9.860194	9.652887	10.29508	10.880677	10.109759	9.797635	9.6235895	9.521514	9.460303	9.422463	9.397975	9.381108	9.368608	9.358644	9.3502035
max_verstap	9.94284	9.945194	9.409912	9.12721	8.978624	8.898174	8.855175	8.833352	8.823504	8.820321	8.8207035	8.822829	8.8256235	8.828446	8.830918	9.763958	11.068215
sainz	10.134467	10.057729	9.685452	9.490819	9.386884	9.32606	9.288066	9.262734	9.244481	9.230163	9.21801	9.207044	9.19674	9.186824	9.177171	9.167726	9.158469
ocon	10.360775	10.225587	9.823066	9.61738	9.50721	9.443447	9.404929	9.380776	9.3649435	9.354009	9.346031	9.33992	9.335066	9.331128	10.079632	10.8686	9.986222
stroll	10.336966	10.223544	9.815226	9.607026	9.496308	9.432479	9.393927	9.369617	9.353435	9.341918	9.333105	9.325907	9.319724	9.314241	9.309293	9.304795	9.300706
gasly	10.247558	10.241831	9.850135	9.646085	9.538001	9.476396	9.439813	9.417348	9.402985	9.393334	9.386481	9.381351	9.377348	9.374141	9.371541	9.369439	10.139894
leclerc	9.936349	9.89965	9.552265	9.371842	9.272202	9.213248	9.177507	9.155417	9.141412	9.132236	9.125989	9.12156	9.118283	9.115754	9.113713	9.111983	9.94999
norris	9.985477	9.982241	9.648193	9.462797	9.359029	9.297904	9.260373	9.236155	9.219514	9.207232	9.1975	9.1893015	9.182074	9.991611	10.660956	9.838128	9.5582
russell	9.940876	10.003676	9.638799	9.439382	9.331158	9.26884	9.231402	9.208011	9.192708	9.182151	9.174447	9.168528	9.1637945	9.932925	10.687099	9.828942	9.535867
albon	10.371836	10.157043	9.78452	9.595321	9.494243	9.435195	9.398719	9.374948	9.358431	9.346095	9.336209	9.327816	9.320392	9.313659	9.307475	10.063058	10.736375
tsunoda	10.196336	10.146868	9.743388	9.535442	9.424809	9.361369	9.323425	9.299907	9.284697	9.274346	9.26691	9.261299	9.256913	9.253418	9.250632	9.248445	9.24679
zhou	10.606307	10.138489	9.788416	9.613305	9.519411	9.463636	9.428103	9.403802	9.385753	9.371143	9.358382	9.346593	10.088069	10.637058	9.915296	9.663701	9.5253
piastri	10.223572	10.079272	9.730218	9.543774	9.443177	9.385077	9.349621	9.326708	9.310854	9.299	9.289436	9.281216	10.109998	10.703784	9.903937	9.634358	9.486821
sargeant	9.28239	8.771414	8.669437	8.602786	8.557865	8.524151	8.496557	8.472347	8.450013	8.428738	8.408106	8.3879385	8.368192	8.348906	8.7038355	8.889254	8.64858

Figure 2: Snapshot of the Predictions from the 2023 Abu Dhabi Grand Prix