F1 Race Winner & Strategy Prediction using Machine Learning

Khushal Yadav
IIIT Delhi
New Delhi, Delhi
khushal22249@iiitd.ac.in

Naman Birla
IIIT Delhi
New Delhi, Delhi
naman22310@iiitd.ac.in

Omansh Arora
IIIT Delhi
New Delhi, Delhi
omansh22342@iiitd.ac.in

Abstract

Formula 1 (F1) racing is a highly competitive sport that requires teams to make strategic decisions throughout the race to optimize their performance and increase their chances of winning the race. This paper aims to develop a machine learning (ML) model capable of predicting F1 race winners and providing optimal strategic decisions and recommendations, such as number of pit stops and tire selections, to improve their odds of winning. The MLP model is trained using multi-class cross-entropy loss and the Adam optimizer with a learning rate of 0.01, enabling it to effectively differentiate strategies and improve prediction accuracy. In addition, we plan to fine-tune hyperparameters such as learning rate, batch size, and the architecture of the model, including the number of layers and neurons, to identify the optimal configuration. Finally, our exploration of evaluation metrics like accuracy, F1 score (macro and micro), precision, and recall deepened our understanding of their relevance in binary classification problems.

1. Introduction

Formula 1 (F1) racing is a highly competitive sport that requires teams to make strategic decisions throughout the race to optimize their performance and increase their chances of winning. Factors such as number of pit stops, tire selection, fuel management, race strategy and driver performance all play a crucial role in determining the winner. Recently in 2018, F1 partnered with Amazon Web Services (AWS) to revolutionize the sport of Formula 1 racing, by enabling the harnessing of Machine Learning techniques and High Performance Cloud Computing (HPCC) [8].

Developing a tool that can predict race winners and provide strategic recommendations could be invaluable for F1 teams [6] [5] [3].

This project report aims to use Machine Learning (ML) techniques to develop a model which predicts F1 race winners and provides optimal strategic decisions and recommendations, such as number of pit stops and tire selections, to improve there chances of winning the race. Also, determining the winner of the race also comes in as another application we can exploit with the use ML [5] [3].

2. Literature Survey

Recent studies have investigated the application of machine learning (ML) and statistical techniques in predicting Formula 1 (F1) race outcomes.

Van Kesteren and Bergkamp in 2022 employed a regression model, finding that 86% of race results are influenced by the constructor, with only 14% attributed to driver skill, underscoring the critical role of car performance [4]. Similarly, Bell et al. in 2016 used random-coefficient models to highlight that team performance is more decisive than driver skill, though driver influence becomes significant in challenging conditions such as wet races [1].

Thabtah and Bunker in 2019 demonstrated the efficacy of Artificial Neural Networks (ANN) in sports predictions, where they outperformed expert predictions. Their work emphasized the importance of feature selection and model validation for achieving high prediction accuracy [2]. Sicoie in 2022 extended this by using Support Vector Machines (SVM), Random Forest, and Gradient Boosting, leveraging High-Performance Cloud Computing (HPCC) for real-time data processing, which improved predictions of race winners and championship standings [8].

Addressing the complexities of sports predictions, including factors like pit stops, several studies have utilized ensemble learning and cross-validation techniques to ensure model robustness and prevent overfitting [8] [2]. These methods, combined with real-time telemetry data and historical race performance, form a strong foundation for accurate F1 race outcome predictions and strategic insights.

3. Dataset Details

The dataset is obtained from Kaggle [7] and consists of all F1 data from 1950-2024. The dataset includes files corresponding to drivers, constructors, qualifying, circuits, lap times, pit stops, and results of all F1 races over the years. The dataset consists of data compiled from Ergast [9], which is a compilation of F1 data.

The dataset contains details about **79** circuits where F1 has conducted its races, **215** constructors, and **860** drivers who participated in the **1144** races conducted since 1950. The data also consists of results of the races, having **26524** entries consisting of

every driver of every race, giving details about their points and positions attained in each and every race they participated in since 1950, along with the number of pitstops taken by each driver and their duration in milliseconds and the number of laps completed by the drivers in the races with the time taken to complete each lap. The dataset is quite detailed and large. We also have the details of the points attained by the constructors and drivers in **1132** of the total **1144** races conducted. The dataset will be used to analyze the performance details of various constructors and drivers over the years on specific circuits to develop machine learning models for the prediction of winners.

The data was pre-processed initially to make it fit for model training. The columns that were not required for model training, like the URL, car number, or string entries, were dropped from all the files. The dataset was mostly complete with missing values in columns that were not considered for model training like the time of race or date of race. Several columns required conversion to appropriate data types. For instance, time-related fields (lap times, pit stop durations) stored as strings were converted into numerical formats like milliseconds for uniformity and ease of comparison. The positions in the results were converted to a Since every race has only one winner and other non-winners, the dataset becomes quite imbalanced in terms of the output variables, with a ratio of around 25:1 for non-winners to winners. Certain data files containing continuous data, like the pitstop durations or lap times, were measured on different scales. To standardize the data for algorithms sensitive to feature scaling, the variables were normalized using standardization, ensuring that no variable solely determines the outcome of the race.. We then split the data into an 80-20 ratio for training and testing to assess bias and variance using scikit-learn.binary output based on whether the driver was the winner. The data for laps and pitstop durations was visualized through various plots like Box plots, histograms, and violin plots to detect outliers. Outliers were detected using boxplots and the **IQR** method where values falling below the lower bound (Q1 - 1.5 x IQR) or above the upper bound (Q3 + 1.5 x IQR), where Q1 is the value below which 25% of the data distribution lies, whereas Q3 is the value below which 75% of the data distribution lies, were considered outliers. These extreme values, particularly in features such as lap times, pit stop durations, and race completion times, were removed to prevent them from skewing the results. Since every race has only one winner and other non-winners, the dataset becomes quite imbalanced in terms of the output variables, with a ratio of around 25:1 for non-winners to winners. Certain data files containing continuous data, like the pitstop durations or lap times, were measured on different scales. To standardize the data for algorithms sensitive to feature scaling, the variables were normalized using standardization, ensuring that no variable solely determines the outcome of the race.. We then split the data into an 80-20 ratio for training and testing to assess bias and variance using scikit-learn.

3.1. Preprocessing for Pit strategy predictor Model

We merged various CSV files to extract important features for predicting pit stops. This was followed by exploratory data analysis (EDA) to remove irrelevant features.

For categorical data, we applied one-hot encoding. For continuous data, we performed standard scaling.

Next, we focused on assessing only the pit stop strategies used by teams that scored high points in the race. After narrowing down the data range, we utilized a correlation matrix to further evaluate the features.

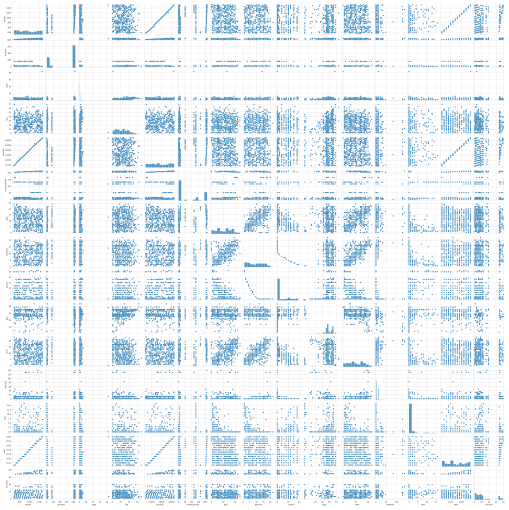

Figure 1. Pair Plot for the dataset of Pit strategy Model

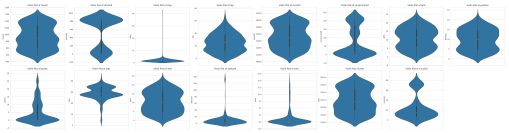

Figure 2. Violin Plot for the dataset of Pit strategy Model

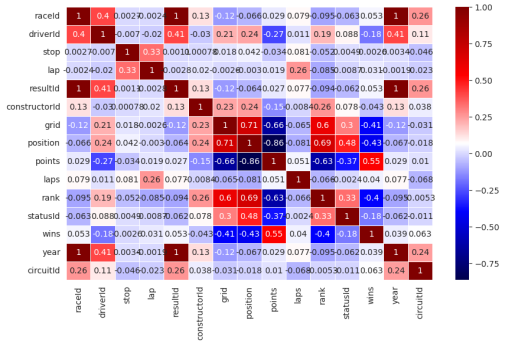

Figure 3. Correlation Plot for the dataset of Pit strategy Model

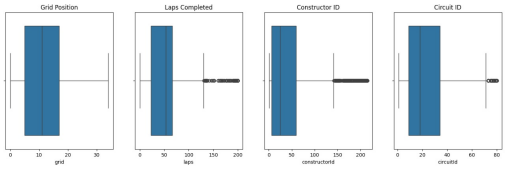

Figure 4. Box plot and outlier detection for winner-predictor model

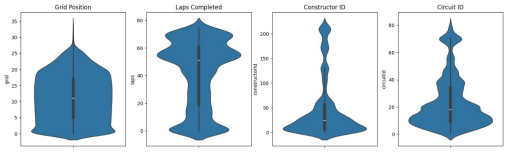

Figure 5. Violin Plot for the dataset of winner prediction model

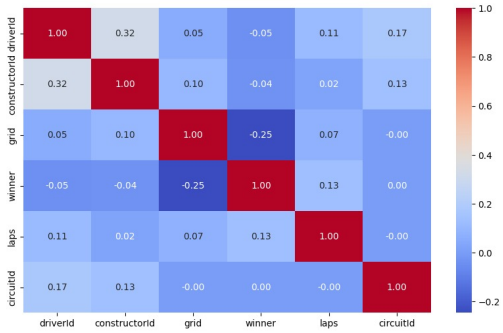

Figure 6. Correlation plot for dataset of winner prediction model

4. Methodology

4.1. Pipeline

Our approach employs a sophisticated two-stage pipeline aimed at accurately predicting race winners by identifying the optimal pit strategy. This comprehensive methodology integrates various influential factors, including track characteristics, race ID, and other relevant features, which will be elaborated upon in subsequent sections. In the first stage, we focus on the Pit Stop Strategy Predictor, which is crucial for determining the ideal timing and approach for pit stops during a race. This stage leverages a Multi-Layer Perceptron (MLP) architecture, specifically designed to output scores ranging from 10 to 25, effectively ranking different pit strategies. Our tech stack is rooted in PyTorch and its sub-modules, allowing for a flexible and robust development environment

4.2. Pit Strategy Predictor Model

Pit Stop Strategy Predictor For predicting the ideal pit stop strategy, we are utilizing a Multi-Layer Perceptron (MLP) designed to score points ranging from 10 to 25. Our tech stack includes PyTorch and its sub-modules.

4.2.1 Architecture

The architecture of our Multi-Layer Perceptron (MLP) comprises six layers, including one input layer, four hidden layers, and one output layer.

4.2.2 Input Layer

The input layer accepts multiple features for predicting optimal pit stop strategies, including Circuit ID, Race ID, Points, Pit Lap, Total Race Laps, and Final Position.

4.2.3 Hidden Layers

Four hidden layers facilitate feature transformation and pattern learning, utilizing various neuron counts and the ELU activation function to introduce non-linearity and enhance gradient flow.

4.2.4 Output Layer

The output layer produces a range of predicted pit strategies as a multi-class classification, with each class representing a unique strategy based on timing and tire changes.

4.2.5 Training Methodolgy

The model is trained using multi-class cross-entropy loss and the Adam optimizer with a learning rate of 0.01, enabling it to effectively differentiate strategies and improve prediction accuracy.

4.3. Winner-predictor model

The model is expected to give a winner prediction given the circuit on which the race is being held, the total laps of the race, the drivers participating in the race, and their grid positions.

4.3.1 Input Data transformation

In order to attain the configuration that can be used to train the model, we convert the categorical data into **one-hot encoded label**. A one-hot encoded label represents each unique category as a binary vector, where only one element is ”1” (indicating the presence of that category), and the rest are ”0”. For example, if the categorical variable is ”constructor” with categories like Ferrari, Mercedes, and Red Bull, each constructor will be represented as a separate binary feature. This transformation ensures that categorical variables are converted into a numerical format that machine learning models can interpret without implying any ordinal relationship between the categories. We construct the one-hot encoded labels of constructors, drivers and circuits in order to represent the presence of different drivers and/or circuits.

4.3.2 Logistic Regression

We observed the correlation between data, and they displayed a weak correlation with respect to each other. The correlation with the dependent variable was quite decent; for example, the grid position showed a 0.30 negative correlation with winner-prediction. Since we needed a binary classification and other factors seemed to settle well, we chose to apply Logistic regression. Now, since the dataset is quite imbalanced with respect to the winner class and non-winner class, we used the weighted logistic regression, which assigns weights to the classes in inverse proportions to their frequencies, ensuring that the rare event of winning a race is adequately represented during the learning process. This helps in achieving better precision and recall for the minority class and prevents the model from being overwhelmed by the majority class, which improves overall performance on imbalanced data.

5. Results

Metric for Pit Stop Predictor Model using MLP: Validation Loss: 0.9739, Macro Accuracy: 0.8424, Micro Accuracy: 0.8424, R^2: 0.7432, F1 Macro: 0.4958, F1 Micro: 0.8424 Metric for Pit Stop Predictor Model using SVM Training Accuracy: 0.8867, Precision: 0.6876, Recall: 0.5958, F1: 0.6310 Test Accuracy: 0.8199, Precision: 0.4673, Recall: 0.4404, F1: 0.4497.
Metric for Winner-predictor model using logistic regression: Training accuracy: 0.88, precision: 0.22, recall: 0.96, F1-score: 0.36 Testing accuracy: 0.87, precision: 0.21, recall: 0.87, F1-Score: 0.34

6. Conclusion

6.1. Pit Stop Count Predictor Model

The Multi-Layer Perceptron (MLP) demonstrates superior performance compared to the Support Vector Machine (SVM) in terms of overall accuracy, exhibiting enhanced generalization to the validation set as evidenced by comparable Macro and Micro Accuracy, R^2 score, and F1 Micro score. Nonetheless, the MLP's F1 Macro score indicates a degree of imbalance in performance across various classes, which warrants further investigation and improvement.

In contrast, the SVM displays elevated training accuracy but suffers a marked decline in performance when evaluated on unseen data, particularly in metrics related to precision, recall, and the F1 score. This phenomenon suggests that the SVM is prone to overfitting, thereby rendering it less reliable for the given problem.

6.2. Winner-predictor model

Even though the logistic regression results in an accuracy of 87%, accuracy is not an appropriate measure because of class imbalance. It shows a high recall of 0.87 but a low precision of 0.21 leading to an overall f1-score of 0.34. It implies that the model correctly predicts 87% of the winners classifies non-winners as winners 79% of the time. herefore, while recall is strong, improving precision is necessary for a better balance between identifying actual winners and minimizing false positives. The model shows similar test and train results implying it is able to generalize well on unseen dataset.

6.3. Learnings

We learned about data piping, exploratory data analysis (EDA), data cleaning, and merging, which enhanced our skills with libraries like Matplotlib, PyTorch, and Scikit-learn, along with data processing using Pandas. Additionally, we collaborated on machine learning projects using Git. During our exploration of handling imbalanced data, we faced challenges that were particularly evident in the F1 macro scores, emphasizing the need to assess performance across all classes rather than focusing solely on accuracy. Implementing the Exponential Linear Unit (ELU) activation function in the MLP model provided insights into how different activation functions can affect convergence and performance in deep learning. We also compared the performance of MLP and Support Vector Machine (SVM) models, gaining a clearer understanding of the strengths and weaknesses of deep learning compared to traditional machine learning techniques. Fine-tuning hyperparameters in the MLP demonstrated the importance of model architecture on validation loss and accuracy. Finally, our exploration of evaluation metrics like accuracy, F1 score (macro and micro), precision, and recall deepened our understanding of their relevance in multiclass classification problems. For logistic regression, we learned how to deal with imbalanced datasets using the concept of assigning class weights. Along with that, we also learnt the relation of correlation matrix with logistic regression, meaning that the logistic regression should be employed if a decent correlation with dependant variable and weak correlation amongst independant variables.

6.4. Work Remaining

We plan to continue improving the MLP model to enhance its generalization capabilities and performance across all classes, with particular attention to addressing the class imbalance reflected in the F1 Macro score. To achieve this, we will explore various strategies. First, we will investigate regularization techniques, specifically L2 regularization, which can help reduce overfitting by penalizing large weights and promoting model simplicity. In addition, we will fine-tune hyperparameters such as learning rate, batch size, and the architecture of the model, including the number of layers and neurons, to identify the optimal configuration. We will also implement techniques like early stopping and dropout to improve the robustness of the model, thereby reducing the chances of overfitting and improving performance on unseen data. Furthermore, we will consider using data augmentation or class balancing techniques to ensure consistent performance across all classes, aiming to improve the F1 Macro score. By executing these strategies, we aim to further refine the MLP model and enhance its overall predictive efficacy. For the winner predictor model, we will incorporate the pit-stop model and merge other datasets for better predictions. We will try to incorporate other models and try to improve the f1-score and will evaluate the best model by using K-Fold cross-validation or using an ensemble of models.

6.5. Team Work

Our team collaborated effectively, ensuring strong coordination and mutual support in enhancing each other's contributions throughout the project. Data gathering was a collective endeavor. Omansh played a pivotal role by conducting research, analyzing relevant literature, and summarizing key methodologies that we could implement followed by assistance in labeling and merging the dataset. Khushal focused on developing the pit stop prediction model, while Naman concentrated on creating the winners prediction model. The contributions of each member were vital to the overall success of the project.

References

[1] Andrew Bell, Philip Smith, Clive Sabel, and Kelvyn Jones. Formula for success: Multilevel modelling of formula one driver and constructor performance, 1950-2014, 2016. Unpublished manuscript. 1

[2] Rory Bunker and Fadi Thabtah. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1):27–33, 2017. 1

[3] Mohamed Hanafy and Ruixing Ming. Machine learning approaches for auto insurance big data. *Risks*, 9(2):42, 2021. Open Access Article. 1

[4] Erik-Jan van Kesteren and Tom Bergkamp. Bayesian analysis of formula one race results: disentangling driver skill and constructor advantage. *Journal of Quantitative Analysis in Sports*, 19(4):273–293, 2023. Open Access Published by De Gruyter July 25, 2023. 1

[5] Ł. Neumann, R. M. Nowak, R. Okuniewski, and P. Wawrzyński. Machine learning-based predictions of customers' decisions in car insurance. *Applied Artificial Intelligence*, 33(9):817–828, 2019. 1

[6] Musa Oytun, Cevdet Tinazci, Boran Sekeroglu, Caner Acikada, and Hasan Ulas Yavuz. Performance prediction and evaluation in female handball players using machine learning models. *IEEE Access*, 8:116321–116335, 2020. 1

[7] Rohan Rao. Formula 1 world championship (1950-2020). https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020, 2020. Accessed: 2024-10-23. 1

[8] Sicoie. Machine learning framework for formula 1 race winner and championship standings predictor, 2022. 1

[9] Ergast Development Team. Ergast motor racing data api. http://ergast.com/mrd/, 2024. Accessed: 2024-10-23. 1