Binary Classification of Formula 1 World Championship Dataset using Random Forest and Neural Networks

Baanee Singh & Shaan Sidhu

Abstract

This project explores the use of machine learning models to predict whether a Formula 1 driver will achieve a podium finish (top 3) based on structured historical race data. Using the Formula 1 World Championships dataset (1950-2020) from Kaggle, we engineered a binary classification task leveraging features such as grid position, constructor affiliation, and race metadata. After data cleaning, feature encoding, and target engineering, we trained and evaluated both a neural network and a Random Forest classifier using a temporal data split, training on data before 2018 and testing on races from 2018 onward.

The neural network achieved an accuracy of 89.65%, with a precision of 0.7125, recall of 0.5213, F1 score of 0.6021, and an AUC score of 0.9172. The Random Forest classifier delivered slightly lower accuracy of 87.9%, but showed stronger performance in classifying podium finishes with a recall of 0.78 and an F1 score of 0.66 for the positive class. Its overall AUC score was 0.9163, indicating robust model discrimination. These results suggest that historical race data can be effectively leveraged to predict top-tier performance, with potential implications for race strategy development and data-driven fan engagement platforms.

Introduction

Predicting the performance outcomes of Formula 1 racing presents a practical application of race strategy, team decision-making, and fan engagement. This project will explore the practicality of predicting whether a driver will finish on the podium (top 3) in a race using structured historical data. Framed as a binary classification problem, the study leverages key performance indicators such as qualifying position, team affiliation, and race-specific metadata.

We utilized the publicly available Formula 1 World Championship 1950–2020 dataset from Kaggle, which contains detailed information on races, drivers, teams, results, and other performance metrics. Data preparation involved merging multiple tables via shared identifiers, handling missing values, removing duplicates, and engineering a binary target variable (podium). Categorical variables were encoded numerically for model compatibility.

To evaluate real-world predictive performance, we employed a temporal train-test split: data from races prior to 2018 was used for training and validation, while races from 2018 onward served as a test set. This approach simulates a deployment scenario where the model is applied to future, unseen data.

Methodology

We chose to do two different machine learning models on our data: Random Forest Model and a Neural Network.

For our neural network model we did a feedforward, fully connected neural network using pytorch. We extended the built in pytorch nn.Module to customize it for our data. Our model consists of 64 neurons in the input layer, 32 neurons in the second layer, 16 neurons in the third layer and 1 neuron in the output layer. The architecture of the neural network was chosen to start with more neurons and then compress down to one as this is a binary classification problem, so binary output is expected from our neural network. It is a common deep learning method to reduce neuron amount by half, going through each layer. The model has a dropout rate of 0.3, which means 30% of the neurons are set to 0, for randomization and to make sure the model doesn't overfit.

Conclusion

Add your information, graphs and images to this section.

Acknowledgements

1. Google Colab User Manual
2. https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020/code
3. In-Class Coding Notebooks & Homeworks of CSci 4521