Predicting Formula 1 Podium Positions

Stanford CS229 Project

Cristian Tavarez
Stanford University
ctavarez@stanford.edu

Joel Sinchi
Stanford University
joel.sinchi@stanford.edu

Zachary Phelps
Stanford University
zphelps@stanford.edu

1 Introduction

Formula 1 racing represents one of the most technologically advanced and strategically complex sports, where outcomes depend on driver skill, car performance, team strategy, circuit characteristics, and weather conditions. Predicting podium finishes presents a challenging machine learning problem with valuable applications for teams planning strategies, broadcasters enhancing viewer engagement, and the growing betting market.

The input to our algorithm is historical F1 race data, including driver characteristics, constructor performance, circuit-specific results, and season progression metrics. We use four distinct models—Logistic Regression, Random Forest, XGBoost, and Neural Networks—to output predicted podium placements (1st, 2nd, and 3rd positions) for each driver in upcoming races.

We employ a rigorous temporal validation methodology, training on pre-2024 seasons and testing exclusively on the 2024 season to simulate real-world prediction scenarios. Additionally, we introduce a domain-specific evaluation metric, the Weighted Podium Score (WPS), to better evaluate prediction performance in the context of F1 racing. Our research addresses key machine learning challenges including class imbalance, temporal dependencies, and the need for interpretable predictions in high-stakes environments.

2 Related Works

The rise of machine learning in sports analytics has significantly improved outcome prediction in complex competitive environments, including Formula 1.

Bunker and Thabtah (2019) highlights the importance of chronological data splits in sports modeling to prevent data leakage and better reflect real-world progression. Similarly, Horvat and Job (2020) shows that neural networks outperform simpler models, achieving up to 85% accuracy in sports prediction, particularly when combined with expert-driven or embedded feature selection methods that integrate selection into the modeling process. These findings suggest that F1-specific factors, like constructor points, track configuration, and driver points, could be essential for podium prediction and underscore the need for chronological data separation when forecasting future seasons.

Ensemble methods have also proven effective. Gu et al. (2019) uses Bagging, Adaboost, and RobustBoost with PCA and SVM to surpass 90% accuracy in hockey prediction. Their success in multi-model approaches supports our use of hybrid methods for F1 forecasting. Meanwhile, van Kesteren and Bergkamp (2023) proposes a Bayesian rank-ordered logit model to separate driver skill from constructor dominance, reinforcing the importance of track conditions and team performance—features. However, while their approach provides clarity, it may not capture nonlinear interactions as effectively as neural networks or ensemble models, which we explore in our work.

Another challenge in F1 forecasting is evaluating how well models predict the top-placed drivers. Sicoiec (2022) uses machine learning methods, SVM, RFR, and GBR to predict the final driver standings, and finds that traditional metrics like R² fail to capture ranking precision and advocate for a top-10-drivers margin of error metric. Learning from this, we incorporate custom evaluation metrics to gain insight into how well our models predict the individual podium positions.

3 Dataset and Features

Our dataset is derived from the 1950-2024 Formula 1 World Championship dataset by Rao (2020), comprising race results, qualifying data, and driver/constructor information. We used seasons through 2023 for training (25,000 driver-race pairs) and reserved the 2024 season for testing (480 driver-race pairs). This temporal split simulates real-world conditions where models predict future events based on historical data. Each instance represents a driver's participation in a specific race, with the target variable being podium achievement.

3.1 Preprocessing

Missing values for historical metrics were replaced with conservative estimates (e.g., 20 for circuit positions, 10 for season positions) to avoid artificial advantages. Numerical features were normalized to zero mean and unit variance, while categorical features (driver_id, constructor_id, circuit_id) were one-hot encoded.

Podium finishes account for only 15% of instances, creating a strong class imbalance. To address this, we applied SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic samples by interpolating between existing podium instances. For a minority-class instance $\mathbf{x}_a$ and one of its closest neighbors $\mathbf{x}_b$, SMOTE generates a synthetic point $\mathbf{x}_{\text{syn}}$ as:

$$\mathbf{x}_{\text{syn}} = \mathbf{x}_a + \lambda\,(\mathbf{x}_b - \mathbf{x}_a),$$

where $\lambda \sim \mathcal{U}(0,1)$. This balances the dataset and improves recall for podium predictions. Additionally, we incorporated class weighting in models supporting it (Logistic Regression, Random Forest, XGBoost) to further mitigate bias toward the majority class penalizing misclassifications of the minority class more heavily during training.

Our final feature set included both raw and engineered features designed to capture driver skill, constructor strength, and recent performance trends. The primary features used were: `raceId`, `year`, `round`, `circuitId`, `driverId`, `constructorId`, `driver_name`, `constructor_name`, `career_wins`, `career_podiums`, `career_titles`, `prev_season_points`, `prev_season_wins`, and `prev_season_position`. Additionally, we engineered expert-driven recent form metrics to captures a driver's momentum: `circuit_avg_position`, `circuit_best_position`, `circuit_races_count`, and driver performance from recent races (`last_5_races_points`, `last_5_races_avg_position`, `last_5_races_count`).

4 Methodology

4.1 Model Selection

4.1.1 Logistic Regression

We chose logistic regression as a baseline for its interpretability and efficiency. It estimates the probability of a podium finish using

$$P(Y = 1 \mid X) = \sigma(\theta^\top X).$$

While relatively simple, it serves as a strong benchmark by which we can compare more sophisticated models. To prevent overfitting, we applied L_1 regularization during hyperparameter tuning.

4.1.2 Random Forest

Random Forest was selected to capture complex non-linear relationships and feature interactions that the logistic regression model might miss. This ensemble method creates multiple decision trees on random subsets of data and features, then aggregates their predictions:

$$f(X) = \frac{1}{B} \sum f^b(X)$$

where B is the number of trees and f^b is the prediction of the b-th tree.

4.1.3 XGBoost

XGBoost incrementally strengthens a weak ensemble of trees by iteratively reducing errors. At iteration t, the model adds a new tree $h_t(x)$ to correct the residual errors from the previous model $f^{(t-1)}(x)$:

$$f^{(t)}(x) = f^{(t-1)}(x) + \eta\, h_t(x),$$

where hyperparameter η is the learning rate. XGBoost uses second-order gradient approximations of the logistic loss to efficiently fit $h_t(x)$.

4.1.4 Feed-Forward Network

We selected a Feed-Forward Neural Network because it has the potential to learn complex nonlinear relationships by passing features through multiple layers of transformations and activation functions. Let $\mathbf{x}$ be the input feature vector. In layer ℓ,

$$\mathbf{h}^{(\ell)} = \sigma\big(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\big),$$

where $\mathbf{h}^{(0)} = \mathbf{x}$ and $\sigma(\cdot)$ is the nonlinear activation ReLU function. The final layer outputs a single logit z, which is passed through the sigmoid function for binary classification:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}.$$

We trained the network by minimizing the cross-entropy loss via backpropagation using an Adam optimizer with batch updates.

4.2 Hyperparameter Optimization

We performed rigorous hyperparameter optimization through temporal cross-validation (5 folds) and randomized search with ROC-AUC as the optimization metric. We chose ROC AUC instead of threshold-dependent metrics like accuracy or F1 to ensure confidence scores accurately reflect podium likelihoods and improve position discrimination. This approach respects the chronological nature of Formula 1 data by ensuring models were always trained on past data and evaluated on future data. The optimal hyperparameter configurations for each model were:

Logistic Regression: L_1 regularization (C=0.0107), liblinear solver, max_iter $= 2584$, no class weighting, SMOTE k_neighbors=9

Random Forest: 104 trees, max_depth $= 10$, min_samples_split $= 13$, min_samples_leaf $= 3$, max_features = None, balanced class weighting, bootstrapping, and SMOTE k_neighbors $= 6$

XGBoost: 181 trees, max_depth $= 9$, learning_rate $= 0.0595$, colsample_bytree $= 0.6186$, gamma $= 0.3038$, min_child_weight $= 2$, subsample $= 0.9769$, SMOTE k_neighbors $= 8$

Feed-Foward Network: hidden_dimension=64, learning_rate=0.0001, batch_size=64, epochs=10

All models incorporated SMOTE for addressing class imbalance with model-specific k_neighbor parameters identified during optimization.

5 Experiments and Results

We evaluated all models on the 2024 Formula 1 season data, which was completely held out during training and optimization. For podium position ranking, we used the predicted probabilities from each binary classification model to rank drivers within each race and assigned positions 1-3 to the highest-ranked drivers.

5.1 Evaluation Metrics

Beyond standard classification metrics such as accuracy, precision, recall, and F1-score, we also designed domain-specific metrics to better capture the nuances of our podium prediction task:

Position-Specific Accuracy: The proportion of races where the model correctly identified the driver in each specific podium position (1st, 2nd, and 3rd).

Weighted Podium Score (WPS): A novel metric we developed assigns different weights for correctly predicting each position, reflecting the F1 points awarded for that placement:

$$WPS = (25 \times C_1 + 18 \times C_2 + 15 \times C_3)/(3N)$$

where C_1, C_2, and C_3 are the counts of correctly predicted 1st, 2nd, and 3rd positions respectively, and N is the number of races. This metric recognizes the greater importance of accurately predicting winners versus lower podium positions, mimicking Formula 1's own championship points system.

5.2 Results

Table 1 presents the standard binary classification metrics for each model on our test set (2024 season).

Table 1: Binary Classification Performance Metrics (2024 Season)

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
Logistic Regression	0.8100	0.7075	0.8482	0.7324	0.9134
Random Forest	**0.8998**	**0.7969**	**0.8667**	**0.8251**	0.9309
XGBoost	0.8852	0.7755	0.7724	0.7739	**0.9355**
Feed-Forward Network	0.8309	0.7218	0.8547	0.7521	0.9067

Random Forest demonstrated the strongest overall binary classification performance, while XGBoost showed the best discrimination ability with the highest ROC AUC score of 0.9355. The Feed-Forward Network and Logistic Regression models delivered competitive performance but were consistently outperformed by the ensemble methods. All models achieved strong recall scores for the podium class (above 0.77), indicating they successfully identified the majority of actual podium finishes.

The confusion matrices in Figure 1 reveal key differences in model performance. Random Forest generally exhibited the best balance between precision and recall as significantly reduced false positives while still correctly identifying most podium finishers. XGBoost, despite its strong ROC AUC, struggled more with recall, misclassifying a notable number of podium finishes as off-podium. Both Logistic Regression and the Neural Network suffered from an increased number of false positives whcih led to an over-prediction of podium finishes. These results suggest that ensemble methods, particularly Random Forest, provide the most reliable classification for distinguishing podium contenders from the rest of the grid.

Table 2 shows the domain-specific F1 podium ranking metrics that evaluate how well each model predicts the specific podium positions.

Table 2: Domain-Specific Racing Metrics (2024 F1 Season)

Model	**Position-Specific Accuracy**			**Weighted Podium**
	1st	**2nd**	**3rd**	**Score (WPS)**
Logistic Regression	0.4167	**0.3333**	0.1667	6.3056
Random Forest	0.4583	0.2500	**0.2500**	6.5694
XGBoost	**0.5417**	0.2917	0.1667	**7.0972**
Feed-Forward Network	0.3333	0.2500	0.2500	5.5277

XGBoost clearly demonstrated superior performance in position-specific prediction tasks, correctly predicting the race winner in 54.17% of races (13 out of 24) and achieving the highest WPS. All models showed a consistent pattern of higher accuracy for the 1st position compared to 2nd and 3rd positions which suggests that predicting the race winner is more straightforward than predicting the rest of the podium. This could be because the performance gap between the dominant driver/car combination and the rest of the field is often more pronounced than the differences between the 2nd and 3rd place finishers.

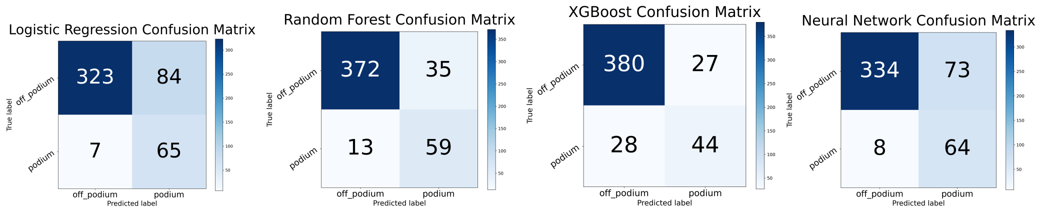

Figure 1: Confusion Matrices for all 4 models

6 Discussion

6.1 Model Comparison

Our results revealed distinct performance patterns across models. Random Forest achieved superior standard classification performance (accuracy: 0.8998, F1 score: 0.8251), while XGBoost excelled in position-specific predictions (race winner accuracy: 54.17%, WPS: 7.0972). This divergence likely stems from their algorithmic differences: Random Forest's bagging approach effectively reduces variance for binary classification, while XGBoost's sequential error correction better captures subtle positional distinctions.

Both ensemble methods outperformed traditional approaches by effectively modeling complex feature interactions and handling mixed data types. Logistic Regression showed surprising effectiveness in predicting second-place finishes (33.33% accuracy) despite its limited expressiveness. The Neural Network, despite its theoretical and proven advantages, failed to surpass the ensemble methods, suggesting that our dataset size (25,000 examples) may be of insufficient size for deep learning approaches without extensive feature engineering.

6.2 Limitations and Challenges

Several limitations affect our prediction system. First, our models cannot account for in-race events such as weather changes, safety cars, or mechanical failures which significantly impact race outcomes and represent a fundamental limitation of pre-race prediction approaches. Second, F1's periodic regulation changes (e.g., the 2022 reset) challenge the temporal validity of predictions. Third, the limited number of races per season (20-24) restricts the sample size for specific driver-constructor-circuit combinations.

Additionally, our probability-based ranking approach does not explicitly model relationships between finishing positions, which may explain the disparity between race winner prediction accuracy (54.17%) and lower podium positions. Finally, our models struggled with predicting occasional podium finishes for midfield teams, indicating difficulties in capturing in-season development trajectories, driver crashes, or car reliability issues.

7 Conclusion / Future Work

Our study demonstrates the feasibility of predicting Formula 1 podium finishes using machine learning techniques. Random Forest achieved the best standard classification performance, while XGBoost excelled at position-specific predictions. Our analysis confirms that ensemble methods effectively capture the complex interactions between team, driver, and circuit-specific features that determine race outcomes. While our approach yields promising results, it remains limited by its inability to account for unpredictable in-race events and dynamic race conditions. Nevertheless, the models provide a solid foundation for predicting F1 results with potential value for teams, broadcasters, and betting markets. Future work should explore: (1) real-time prediction updates incorporating qualifying results and live race data; (2) advanced time series modeling to better capture performance trends; (3) multi-class approaches that directly predict specific finishing positions; (4) integration of external data sources including weather forecasts, tire strategies, regulation overhauls; and (5) probabilistic models that quantify prediction uncertainty. These directions could significantly enhance prediction accuracy and robustness for Formula 1 racing outcomes.

8 Contributions

Zach Phelps: Prepared training/test datasets. Built logistic, random forest, XGboost, and Neural Network models. Conducted training and evaluation. Worked on the Hyperparameter, Experiments and Results, Discussion, and Conclusion sections of the paper.

Cristian Tavarez: Created helper functions to pull data. Worked on the Related Works, Methodology, Discussion, and Conclusion sections of the Paper.

Joel Sinchi: Generated plots on features-to-win and did research to select domain-specific expert-driven features. Worked on the Introduction and Dataset Features, and Conclusion sections of the Paper.

References

Rory P. Bunker and Fadi Thabtah. 2019. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1):27–33.

Wei Gu, Krista Foster, Jennifer Shang, and Lirong Wei. 2019. A game-predicting expert system using big data and machine learning. *Expert Systems with Applications*, 130:293–305.

Tomaž Horvat and Jan Job. 2020. The use of machine learning in sport outcome prediction: A review. *WIREs Data Mining and Knowledge Discovery*, 10:e1380.

Erik-Jan van Kesteren and Tom Bergkamp. 2023. Bayesian analysis of formula one race results: disentangling driver skill and constructor advantage. *Journal of Quantitative Analysis in Sports*, 19(4):273–293.

Rohan Rao. 2020. Formula 1 world championship 1950-2020. Accessed: 2025-03-11.

Horatiu Sicoiec. 2022. Machine learning framework for formula 1 race winner and championship standings predictor.