Predicting Formula 1 Race Winners Using Machine Learning

ECGR 4105 Intro to Machine Learning

Dr. Hamed Tabkhivayghan

Alex Ayerbe
ECE
UNC Charlotte
Charlotte, NC
aayerbe@uncc.edu

Harshil Brinda
ECE
UNC Charlotte
Charlotte, NC
hbrinda@charlotte.edu

Abdirahim Ahmed
ECE
UNC Charlotte
Charlotte, NC
dahmed3@charlotte.edu

Github URL: https://github.com/HeedfulMoss/ML-F1-Prediction-Project

I. Introduction

In recent years, machine learning has revolutionized many industries, offering new insights and efficiencies previously unattainable. Our project, "Predicting Formula 1 Race Winners Using Machine Learning," leverages this transformative technology to analyze and predict outcomes in one of the most technologically advanced and competitive sports in the world—Formula 1 racing.

The motivation behind this project stems from the sport's complex nature, where numerous variables such as driver skill, car performance, team strategies, and even weather conditions can influence race outcomes. By applying machine learning techniques, we aim to develop a predictive model that not only enhances the understanding of key performance indicators but also supports F1 teams in optimizing race strategies and improving overall competitiveness. The broader impact of our project includes aiding teams in decision-making processes and enhancing the spectator experience by providing deeper insights into potential race outcomes,

boosting fan engagement and potential marketing plays.

II. Approach

Our approach to tackling the challenge of predicting Formula 1 race winners involves a meticulous process of data handling and model selection. Initially, we collected extensive historical race data that includes variables such as driver performance, car specifications, team strategies, and race conditions. This dataset underwent a rigorous preprocessing phase to clean and structure the data, involving techniques to handle missing values, normalize features, and encode categorical data effectively.

In the feature selection phase, we employed Principal Component Analysis (PCA) to reduce dimensionality and isolate the most influential features that impact race outcomes. This step not only streamlined our modeling process but also enhanced the interpretability of our results.

For the model development, we explored three different machine learning algorithms: Logistic Regression, Support Vector Machine (SVM), and Fully Connected Neural Network. For logistic regression, used as a baseline model due to its simplicity and efficiency in binary classification tasks. For SVM, it was chosen for its effectiveness in high-dimensional spaces and its capability to model non-linear decision boundaries thanks to kernel tricks. For the Neural Network, it was implemented to capture complex patterns and interactions in the data, potentially offering higher accuracy at the cost of increased computational complexity.

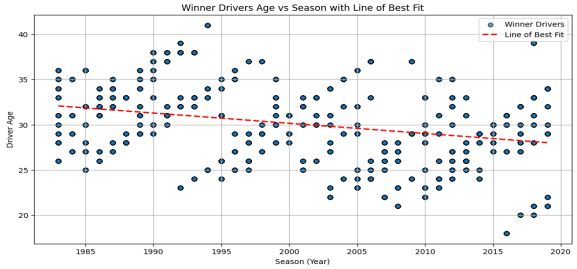

Figure 1.0 Age over year

Each model was trained using an 80%-20% train-test split to ensure they are robust and perform well on unseen data. We evaluated each model's performance through metrics such as accuracy, precision, recall, and F1-score, which helped in fine-tuning the models and selecting the best performer for deployment.

To mitigate overfitting and ensure that our models generalize well, we applied cross-validation techniques across all models. This rigorous validation not only strengthened the reliability of our predictive models but also provided insights into their operational efficacy under different racing scenarios.

III. Dataset and Training Setup

	year	round	circuitId	lat	lng	country	date	url
0	1950	1	silverstone	52.0786	-1.01694	UK	1950-05-13	http://en.wikipedia.org/wiki/1950_British_Gran...
1	1950	2	monaco	43.7347	7.42056	Monaco	1950-05-21	http://en.wikipedia.org/wiki/1950_Monaco_Grand...
2	1950	3	indianapolis	39.7950	-86.23470	USA	1950-05-30	http://en.wikipedia.org/wiki/1950_Indianapolis...
3	1950	4	bremgarten	46.9589	7.40194	Switzerland	1950-06-04	http://en.wikipedia.org/wiki/1950_Swiss_Grand_...
4	1950	5	spa	50.4372	5.97139	Belgium	1950-06-18	http://en.wikipedia.org/wiki/1950_Belgian_Gran...
...	...	...	...	...	...	...	...	...
1120	2024	20	rodriguez	19.4042	-99.09070	Mexico	2024-10-27	https://en.wikipedia.org/wiki/2024_Mexico_City...
1121	2024	21	interlagos	-23.7036	-46.69970	Brazil	2024-11-03	https://en.wikipedia.org/wiki/2024_S%C3%A3o_Pa...
1122	2024	22	vegas	36.1147	-115.17300	United States	2024-11-23	https://en.wikipedia.org/wiki/2024_Las_Vegas_G...
1123	2024	23	losail	25.4900	51.45420	Qatar	2024-12-01	https://en.wikipedia.org/wiki/2024_Qatar_Grand...
1124	2024	24	yas_marina	24.4672	54.60310	UAE	2024-12-08	https://en.wikipedia.org/wiki/2024_Abu_Dhabi_G...

Figure 1.1 Races dataframe1.

	year	round	circuitId	driver	dateOfBirth	nationality	constructor	grid	time	status	points	position
0	1950	1	silverstone	farina	1906-10-30	Italian	Alfa Romeo	1	2:13:23.6	Finished	9.0	1
1	1950	1	silverstone	fagioli	1898-06-09	Italian	Alfa Romeo	2	+2.6	Finished	6.0	2
2	1950	1	silverstone	reg_parnell	1911-07-02	British	Alfa Romeo	4	+52.0	Finished	4.0	3
3	1950	1	silverstone	cabantous	1904-10-08	French	Talbot-Lago	6	\N	+2 Laps	3.0	4
4	1950	1	silverstone	rosier	1905-11-05	French	Talbot-Lago	9	\N	+2 Laps	2.0	5
...	...	...	...	...	...	...	...	...	...	...	...	...
26514	2024	12	silverstone	ocon	1996-09-17	French	Alpine F1 Team	18	\N	+2 Laps	0.0	16
26515	2024	12	silverstone	perez	1990-01-26	Mexican	Red Bull	0	\N	+2 Laps	0.0	17
26516	2024	12	silverstone	zhou	1999-05-30	Chinese	Sauber	14	\N	+2 Laps	0.0	18
26517	2024	12	silverstone	russell	1998-02-15	British	Mercedes	1	\N	Water pressure	0.0	\N
26518	2024	12	silverstone	gasly	1996-02-07	French	Alpine F1 Team	19	\N	Gearbox	0.0	\N

Figure 1.2 Results dataframe2.

	year	round	driverRef	points	wins	position
0	1950	1	farina	9.0	1	1
1	1950	1	fagioli	6.0	0	2
2	1950	1	reg_parnell	4.0	0	3
3	1950	1	cabantous	3.0	0	4
4	1950	1	rosier	2.0	0	5
...	...	...	...	...	...	...
34590	2024	12	albon	4.0	0	17
34591	2024	12	ocon	3.0	0	18
34592	2024	12	zhou	0.0	0	19
34593	2024	12	sargeant	0.0	0	20
34594	2024	12	bottas	0.0	0	21

Figure 1.3 Driver Standing dataframe3.

	year	round	constructor	constructor_points	constructor_wins	constructor_standings_
0	1958	1	Cooper	8.0	1	
1	1958	1	Ferrari	6.0	0	
2	1958	1	Maserati	3.0	0	
3	1958	2	Cooper	16.0	2	
4	1958	2	Ferrari	12.0	0	
...	...	...	...	...	...	
13266	2024	12	RB F1 Team	31.0	0	
13267	2024	12	Haas F1 Team	27.0	0	
13268	2024	12	Alpine F1 Team	9.0	0	
13269	2024	12	Williams	4.0	0	
13270	2024	12	Sauber	0.0	0	

Figure 1.4 Constructor Standing dataframe4.

	grid	constructor	year	round	driverRef	q
0	1	McLaren	2008	1	hamilton	1:26.714
1	2	BMW Sauber	2008	1	kubica	1:26.869
2	3	McLaren	2008	1	kovalainen	1:27.079
3	4	Ferrari	2008	1	massa	1:27.178
4	5	BMW Sauber	2008	1	heidfeld	1:27.236
...	...	...	...	...	...	...

Figure 1.5 Qualifying dataframe5.

	year	round	circuitId	date	url	weather_warm	weather_cold	weather_dry	weather_wet	weather_cloudy
0	1950	1	silverstone	1950-05-13	http://en.wikipedia.org/wiki/1950_British_Gran...	1	0	1	0	0
1	1950	2	monaco	1950-05-21	http://en.wikipedia.org/wiki/1950_Monaco_Grand...	0	0	0	0	0
2	1950	3	indianapolis	1950-05-30	http://en.wikipedia.org/wiki/1950_Indianapolis...	0	0	0	0	0
3	1950	4	bremgarten	1950-06-04	http://en.wikipedia.org/wiki/1950_Swiss_Grand_...	1	0	1	0	0
4	1950	5	spa	1950-06-18	http://en.wikipedia.org/wiki/1950_Belgian_Gran...	1	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...
1120	2024	20	rodriguez	2024-10-27	https://en.wikipedia.org/wiki/2024_Mexico_City...	0	0	0	0	1
1121	2024	21	interlagos	2024-11-03	https://en.wikipedia.org/wiki/2024_S%C3%A3o_Pa...	0	0	0	1	0
1122	2024	22	vegas	2024-11-23	https://en.wikipedia.org/wiki/2024_Las_Vegas_G...	1	0	0	0	0
1123	2024	23	losail	2024-12-01	https://en.wikipedia.org/wiki/2024_Qatar_Grand...	1	0	0	0	0
1124	2024	24	yas_marina	2024-12-08	https://en.wikipedia.org/wiki/2024_Abu_Dhabi_G...	1	0	0	0	0

Figure 1.6 Weather dataframe6.

IV. Results and Analysis

In our exploration of predicting Formula 1 race outcomes using machine learning, the fully connected neural network played a critical role. We experimented with three different architectural configurations to evaluate the effectiveness of each in handling the complexity of our dataset and achieving desirable generalization capabilities. The analysis focused on comparing training and validation losses and accuracies, along with the implications of architectural adjustments on the classification testing accuracy.

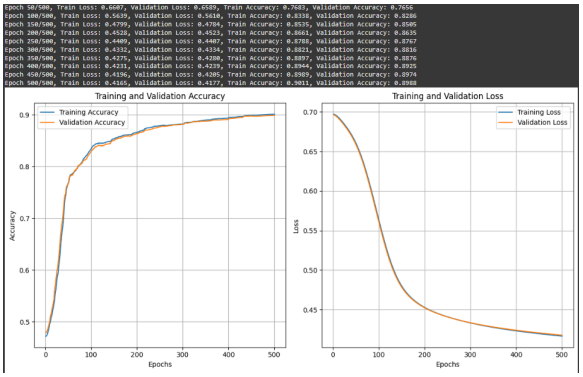

Figure 2.1 Training and Validation for FNN

The initial setup of our neural network consisted of multiple layers with varying neuron counts (75, 25, 50, 10) integrated with ReLU activation functions. The model culminated in an output layer of two neurons with a Softmax function for probability determination. This configuration achieved a training accuracy of 88.67% and a validation accuracy of 87.65%, which were promising. However, the classification testing accuracy was approximately 53%, suggesting that the model might be overfitting the training data, evident from the high discrepancy between training/validation performance and test performance.

```
Classification Report for nn:
              precision    recall  f1-score   support

           0       0.97      0.85      0.90      2504
           1       0.86      0.97      0.91      2456

    accuracy                           0.91      4960
   macro avg       0.91      0.91      0.91      4960
weighted avg       0.91      0.91      0.91      4960

fnn Test Accuracy: 0.4762
```

Figure 2.2 Classification report for FNN

To address potential overfitting, we reduced the complexity of the network by decreasing the number of neurons in the layers (50, 10) and simplifying the architecture. This configuration produced a lower training loss of 0.4106 and a validation loss of 0.4215, with improved accuracies (training: 91.02%, validation: 89.74%). Despite these improvements, the classification testing accuracy remained largely unchanged, which indicated that while the model became more efficient at learning from the training data, it did not significantly enhance its ability to generalize to new, unseen data.

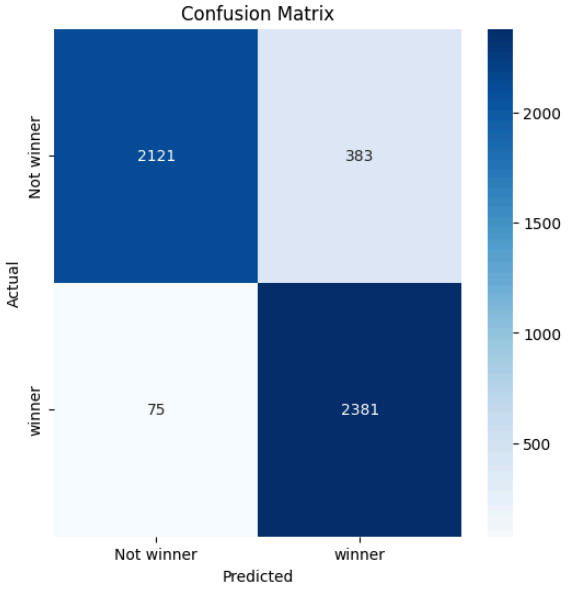

Figure 2.3 Confusion Matrix for FNN

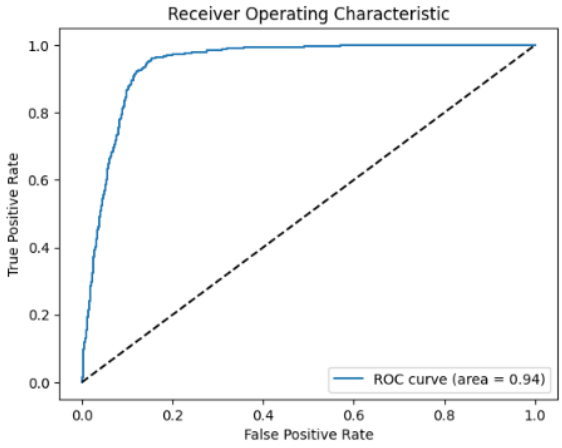

Figure 2.2 ROC Curve for FNN

The final architectural variation introduced dropout layers with a rate of 50% after the first and second hidden layers, aiming to further mitigate overfitting by randomly dropping units during training. This approach adjusted the network's exposure to the full set of features, promoting better generalization. Though this model showed a slight decrease in both training (83.78%) and validation accuracy (85.60%) with increased losses (train: 0.5200, validation: 0.4991), it pointed towards a balance between learning and generalization. Despite these changes, the classification testing accuracy did not show substantial improvement, suggesting a need for further tuning or reconsideration of the dropout rate to optimize performance without losing necessary information.

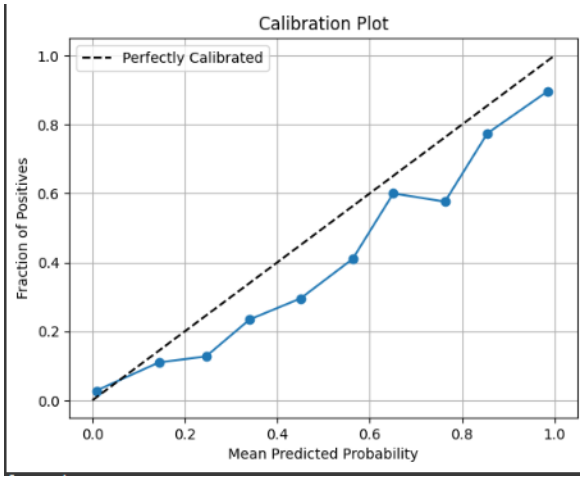

Figure 2.4 Calibration Plot

Through these experiments, we learned that while simplification and regularization can help reduce overfitting, they must be carefully balanced to maintain the model's ability to learn critical patterns from the data. Each configuration offered valuable insights into the dynamics of neural network training and generalization, guiding future adjustments and model selections.

Classification Report for logistic:				
	precision	recall	f1-score	support
0	0.96	1.00	0.98	2654
1	0.48	0.09	0.15	123
accuracy			0.96	2777
macro avg	0.72	0.54	0.56	2777
weighted avg	0.94	0.96	0.94	2777

Logistic Regression Test Accuracy: 0.4762

Figure 3.1 Classification report for Logistic Regression.

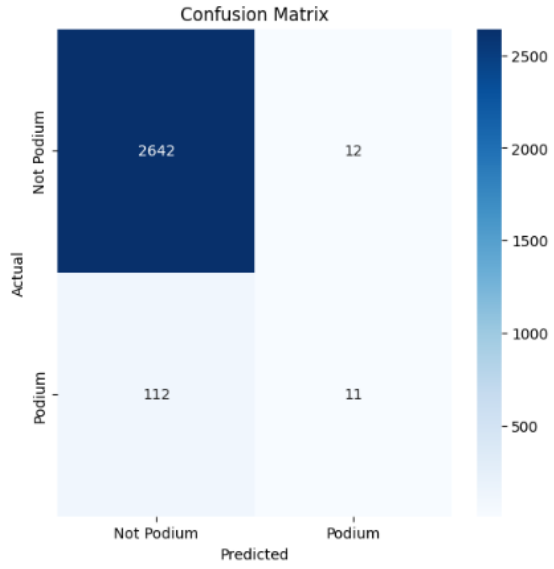

Figure 3.2 Confusion Matrix for Logistic Regression.

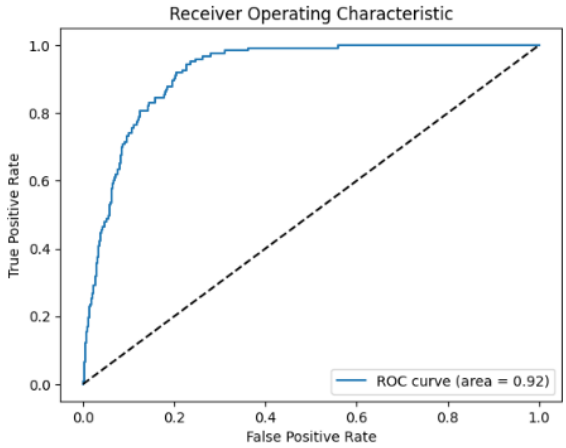

Figure 3.3 ROC Curve for Logistic Regression.

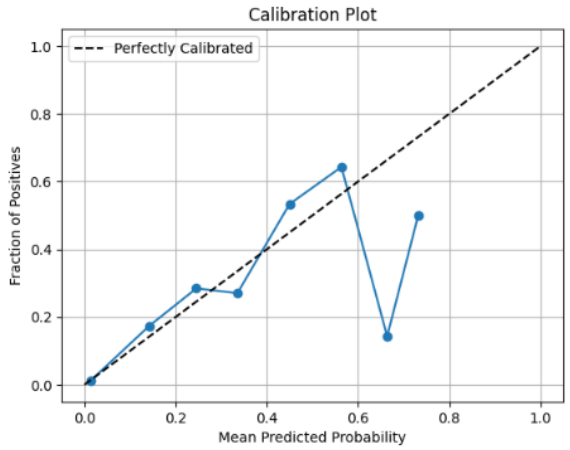

Figure 3.4 Calibration Plot for logistic regression.

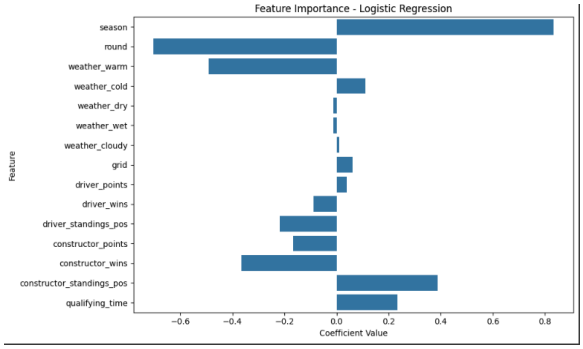

Figure 3.5 Feature Importance Plot for Logistic Regression.

From the feature plot for Figure 3.5, we can analyze that the season a race is taking place has the highest effect on determining race position. Followed by the standings of the constructor for each race.

Classification Report for svm:				
	precision	recall	f1-score	support
0	0.96	1.00	0.98	2654
1	0.80	0.03	0.06	123
accuracy			0.96	2777
macro avg	0.88	0.52	0.52	2777
weighted avg	0.95	0.96	0.94	2777

```
SVM Test Accuracy: 0.4762
```

Figure 4.1 Classification report for SVM

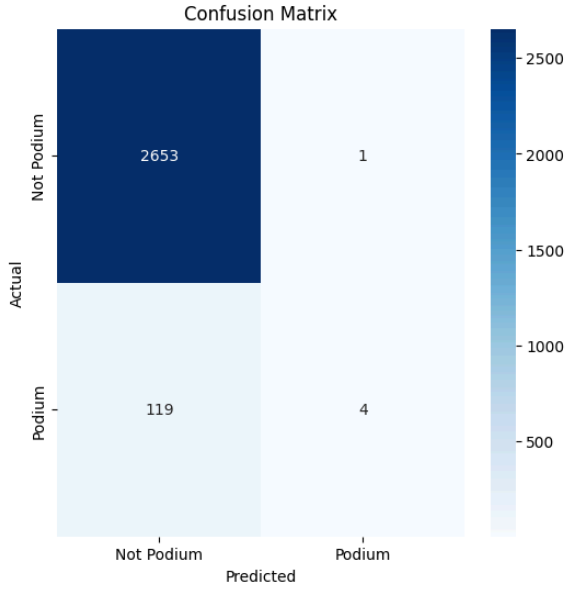

Figure 4.2 Confusion Matrix for SVM

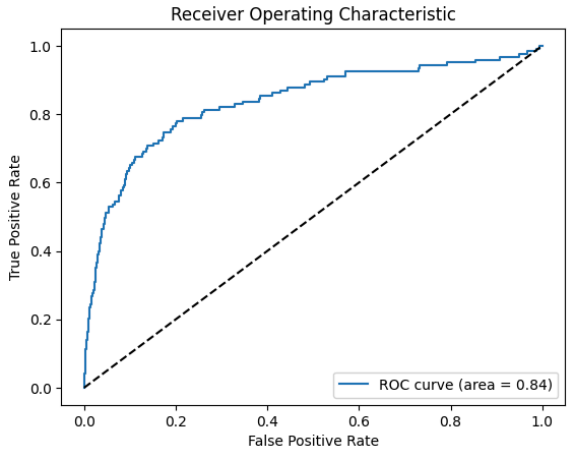

Figure 4.3 ROC Curve for SVM

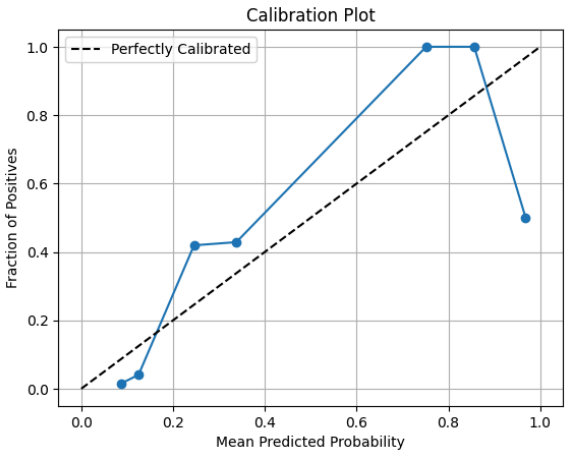

Figure 4.4 Calibration Plot for SVM

V. Conclusion

In conclusion, our analysis revealed that regardless of the model used, overfitting remained a consistent issue. This overfitting was primarily due to the lack of sufficient data for first-position winners. For example, in the case of the neural network, training and validation accuracy were very high, but the loss was also relatively high. When making predictions on the test data, the model was only successful about 50% of the time, indicating poor generalization.

This was further evidenced by the network's tendency to repeatedly predict certain outcomes, such as consistently selecting Hamilton as the winner. The imbalance in the dataset caused the models to rely heavily on historical patterns without capturing the variability in actual outcomes. By focusing on position, we aimed to refine our analysis, but the lack of diverse data for first-place finishes continued to undermine performance. To address this issue, future work should prioritize acquiring more balanced datasets to improve model robustness and reduce overfitting.

In addition, the performance similarity between Logistic Regression and SVM in our experiments highlights an interesting aspect of model behavior under conditions of limited data diversity and prevalent overfitting. Both models, despite their fundamental differences in handling classification tasks, ended up with the same accuracy on unseen data. This outcome could indicate that the linear decision boundaries assumed by both Logistic Regression and linear SVM were sufficient to model the available data's patterns, but not complex enough to generalize beyond the biased training samples.

The convergence in accuracy between these models suggests that the primary challenge lies not in the model selection but rather in the quality and quantity of the training data. As both models showed similar limitations, it emphasizes the need for a more strategic approach to data collection and preprocessing to ensure that future models can learn more representative patterns of the sport, reducing the dependency on dominant trends like specific winners. This approach would involve not only expanding the dataset but also implementing more sophisticated data augmentation and balancing techniques to provide a broader spectrum of training examples for models to learn from.