On The Right Track: Predicting F1 Race Point Finishers using an Ensemble Tree Model Regressor

Leodegario II Lorenzo, Warren Dela Cruz, Jeonne Joseph Ramoso,
Richard Rian, Kyle Christian Uy

March 2024

Abstract
The surge in Formula 1's popularity, fueled by Netflix's 'Drive to Survive', has opened lucrative opportunities. Teams compete for F1 sponsorships reaching up to $75 million annually, while betting enthusiasts engage in a market exceeding $1 billion. Additionally, fantasy gaming is projected to reach $87 billion by 2031 with F1 fantasy included. To help stakeholders capitalize on these trends, our team has developed a regression model that predicts Formula 1 drivers likely to finish in the top ten. This tool enables teams to refine strategies and pinpoint areas for improvement. Bettors can make informed decisions on where to place their bets, and fantasy players can construct winning teams and plan transfers effectively. Among all the tested models, the RandomForestRegressor achieved the highest MAP@K score. Our findings indicate an 89% level of confidence in the model's ability to predict the top ten finishers in any given F1 race. Furthermore, our model incorporates actionable insights using LIME and DiCE explanation methods. It can provide racers with information into why their predictions turned out as they did and suggest improvements for upcoming races.

Keywords: formula one, race predictions, ensemble tree models, explainability

Highlights:

- Prediction features utilize pre-race information and results.
- Each race sees independent predictions for every driver.
- Random Forest Regressor excelled, achieving an 89% MAP@K score.
- LIME aids in identifying key prediction factors.
- DICE facilitates driver performance enhancements over the next five races.

1 Introduction

Formula 1 (F1) racing, established in 1950, is the apex of motorsport, featuring high-speed races, advanced technology, and a global fan base. It showcases the world's top drivers in sophisticated single-seater race crafts, competing in a series of international Grands Prix. The 2022 and 2023 seasons, marked by intense rivalry and captivating moments, notably saw Max Verstappen emerging as a decisive winner for the World Championship.

In the Philippines, the "F1 Club PH" Facebook group exemplifies the sport's growing popularity, offering a platform for fans to share their passion, discuss races, and engage with the community. The Netflix series "Drive to Survive" has further boosted F1's global appeal, drawing in new fans by providing an intimate look into the sport's drama and intricacies.

In this fast-paced world of Formula 1 racing, where the elite of motorsport compete at breakneck speeds, the determination of which drivers are likely to finish in the top ten in any given race is a problem of both strategic importance and significant complexity. The challenge lies in developing a predictive regression model that accurately forecasts these top performers, considering the myriad of variables that influence race outcomes. This endeavor is critical as every point earned contributes significantly to the championship standings, affecting team strategies, driver confidence, and the allocation of resources throughout the season.

The drive to create a regression model predicting top finishers in F1 races is fueled by the model's potential to provide strategic insights for teams vying for a slice of the significant sponsorship revenue which can run up to $75M per year [6], aid sports bettors in a billion-dollar betting industry [5], and enhance F1 fantasy team management in a rapidly growing fantasy sports market projected to expand to $87B per year by 2031 (Industry Growth Report, 2023). This study is poised to not only elevate the strategic aspects of the sport but also tap into the economic benefits for various stakeholders in the Formula 1 ecosystem.

2 Related Works

To establish a benchmark for this study, the team reviewed projects that are related to the intended goal of the research. By comparing the results of similar research, it is possible to understand how this study expands upon and contributes to existing literature. There are around 98 Python notebooks on Kaggle that explore various use cases for the Formula 1 dataset. Most of the implementations are focused on either exploratory data analysis or predictive modeling.

F1 Race Predictions, a project by Robert et. al [9], is designed to predict the winners of each event in the 2019 Formula 1 season, with winners denoted by a '1' and non-winners by a '0' for each race. It utilizes a static dataset that amalgamates details from all seasons since 1989 up to the 2018 season. The modeling incorporates a variety of features, including race weather conditions, track names, driver profiles, team details, race outcomes, as well as aggregate data on team victories and standings. Four different machine learning models were utilized, namely: Neural Network (NN) Regressor, Support Vector Machine (SVM) Regressor, Linear Regression (LR), and Random Forest (RF) Regressor. The results indicate that the RF model achieved the highest accuracy, nearly 70%. For deeper insight, feature importance was extracted, revealing that the starting grid position, points, and driver standings were the most significant factors in making predictions [9].

On the other hand, f1-Predictions, another Formula 1 results prediction project developed by Texas-based data scientist Mark Landry, presents a unique approach. This project, dedicated to his son, serves as a demonstration of the potential of machine learning in various fields such as Formula 1, a shared passion between them. What Landry innovatively introduced in this project was the utilization of rolling calculations in predictions, leveraging race-specific information to forecast outcomes. For example, when predicting results for the British Grand Prix, Landry incorporated past race data from the same circuit. Additionally, the features employed in this project differed significantly. It incorporates driver and team positions from five and ten races prior, along with an output value derived from a custom function that combines weighted qualifying and grid positions, along with the previously mentioned features. This time, the target variable is the final race position for the 2020 British Grand Prix, predicted through a linear regression machine learning model [8].

The two projects vary in features, methods, and prediction targets, providing potential learning points for the team to devise a more novel approach. While mean absolute error and prediction accuracy are commonly used metrics to assess project success, they may not align with the current research's business objectives. Moreover, given the disparities in approach and objectives, the two projects may not serve as suitable benchmarks for comparison. Thus, the team must seek an alternative baseline to gauge success.

3 Data and Methods

3.1 Data Source

3.1.1 Ergast Development Website API

Table 1: Description of Data Features retrieved from Ergast Development Website API

Data Feature	Description
Circuits	Track names and Information
Constructors	Constructor names and Information
Constructor Standings	Constructor standing after every race
Drivers	Driver names and Information
Driver Standings	Driver standing after every race
Lap Times	Driver Lap time for every lap of every race
Qualifying	Driver Qualifying Position for every race
Races	Race information
Results	Result of each driver for every race
Status	Status descriptions

Historical Records of Formula One series are pulled from the Ergast Developer API [4]. This experimental web service provides data for the motor racing sport from the beginning of the 1950s World Championship. The website describes different methods to query the data which included an easy-to-use downloadable zip file containing the zip files of each data table in CSV format. These CSV files (see Table 1) were then loaded into a local sqlite3 database for querying.

3.1.2 F1 Official Website

The team conducted web scraping on the F1 Official website to gather practice session data for the 2022 and 2023 seasons [3]. This process involved iterating through each year and race specified in a dictionary, dynamically forming URLs for each practice session. Utilizing requests for fetching webpage content and BeautifulSoup (bs4) for HTML parsing, the function navigated the DOM structure to locate the data table containing session results. Upon execution, the scraping function produced a comprehensive DataFrame, as depicted in Table 2, encompassing practice session results across the 2022 and 2023 seasons. Subsequently, this DataFrame was incorporated into the local sqlite3 database.

Table 2: Description of Data Features scraped from Official F1 Website

Data Feature	**Description**
Year	Year the session took place
Location	Location or circuit where the session was held
Session	Session number within the event. Formula 1 events typically include multiple practice sessions
Pos	Position or rank of the driver based on their performance in that session
No	Driver's racing number, which is unique to each driver and is used throughout the season
Driver	Driver's name, abbreviated with their last name and a three-letter code
Car	Team and engine supplier or the car model the driver was using for the session
Time	Driver's best lap time in the session
Gap	Time difference between the driver's best lap and the fastest lap in the session. For the driver with the fastest lap, this is typically left blank
Laps	Total number of laps completed by the driver in that session

3.1.3 Wikipedia Race Track Information

Circuit length of each track was also scraped using request and bs4 python packages from each of the circuit's corresponding Wikipedia Page [2]. The length of each circuit was combined with the Circuits data table to create the data table CircuitsNew on the sqlite3 database. This was then used in creating an average lap speed feature of each of the driver's races by dividing the circuit length with the average lap time.

3.2 Dataset Building

After collecting the needed data from the various sources, the working dataset for model training was then constructed. The dataset was limited to the 2022 and 2023 seasons to reflect the rule changes [1] introduced at the start of the 2022 season. These rule changes caused an overhaul of the racecar design, which may heavily create discrepancy with the performance of previous seasons. Drivers that were considered for the dataset should be active for the last 4 races of the 2023 season and should have at least 10 total completed races between the two seasons. Finally, only races where the driver finished were considered to remove unpredictable events such as crashes, engine failures, disqualifications, etc.

The final (set of) datasets included time-series features (see Table 3). Lag features referring to previous five races of the current were created for the following: driver positions, standings, average lap ranking, etc. and as well as constructor standings. These engineered features were created for each driver per race. This resulted to removing the first five races of each driver due to the null values in the lag features.

3.3 Methodology

3.3.1 Evaluation Metrics

To provide context for the chosen metric, it is necessary to briefly outline how the model operates. Put simply, the model predicts the finishing positions of each driver independently using a regressor for a specific race. Following

Table 3: Dataset Dictionary of Features

Data Table	**Description**
positionOrder	True value of the target prediction
1Lag_dsPoints	Driver Points after last race
1Lag_dsPosition	Driver Standing (in the Championship) after last race
1Lag_driverWins	Driver Wins after last race
1Lag_constructorPoints	Constructor Points after last race
1Lag_constructorPosition	Constructor Standing (in the Championship) after last race
1Lag_constructorWins	Constructor Wins after last race
1Lag_aveLapPosition	Driver's Average Lap Position in the last race
1Lag_aveLapSpeed	Driver's Average Lap Speed in the last race
1Lag_fastestLapRank	Driver's ranking for fastest lap in the last race
qualifyingPosition	Driver's Qualifying (Grid) Position for the current race
1LagPosition	Driver's Ending Position in the last race
2LagPosition	Driver's Ending Position in the 2nd to the last race
3LagPosition	Driver's Ending Position in the 3rd to the last race
4LagPosition	Driver's Ending Position in the 4th to the last race
5LagPosition	Driver's Ending Position in the 5th to the last race
1LagQPos	Driver's Qualifying Position in the last race
2LagQPos	Driver's Qualifying Position in the 2nd to the last race
3LagQPos	Driver's Qualifying Position in the 3rd to the last race
4LagQPos	Driver's Qualifying Position in the 4th to the last race
5LagQPos	Driver's Qualifying Position in the 5th to the last race

this prediction, the drivers are sorted from highest to lowest rank to identify which drivers make it into the top ten and which do not.

Out of all the evaluation metrics considered, Average Precision at K (AP@K) emerged as the most suitable measure given the nature of the predictions. The team explored other metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Normalized Discounted Cumulative Gain (nDCG), and Average Recall at K (AR@K). However, the first three metrics were found to be unsuitable, while AR@k is similar to AP@k for our study [7].

AP@K, an evaluation metric designed for ranked data, assesses the proportion of relevant results within the top k predictions while considering their order. Unlike metrics that focus on exact ranking positions, AP@K evaluates accuracy based on the presence of relevant items within the top k results. It penalizes the accuracy score when irrelevant items are ranked highly within the k results. This metric aligns well with the outputs of the model in this research, given its consideration of the ranked nature of final predictions.

To demonstrate, refer to the figure below, which is similar to the example given by Efimov in his article [7].

$$\text{AP@k} = \frac{1}{r}\sum_{i=1}^{k} \text{precision@}i \cdot R_i \tag{1}$$

where r is the number of relevant items and R_i is defined as:

$$R_i = \begin{cases} 1 & \text{if document } i \text{ is relevant} \\ 0 & \text{if document } i \text{ is not relevant} \end{cases}$$

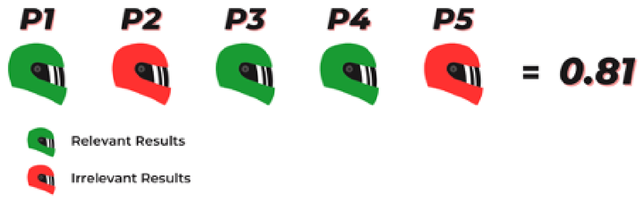

Figure 1: AP@K Sample 1

In the scenario depicted in Figure 1, the computation of AP@K for a hypothetical case results in a value of 0.81, demonstrating the impact of relevant and irrelevant results within the top rankings.

Figure 2: AP@K Sample 2

Contrasting this with the situation in Figure 2, where irrelevant results are ranked even higher at first and third, resulting in an AP@K of 0.53, it is evident that the accuracy is significantly improved.

$$\text{Model MAP@k (score)} = \frac{1}{15}\sum_{i=1}^{15}\left(\frac{i}{\text{rank}_i} \times \text{rel}_i\right)_i \tag{2}$$

The team also employed Mean Average Precision at K (MAP@K) to assess the accuracy of predictions across multiple races. This approach involves calculating AP@K scores for each race and then taking the average. For example, if there are four races, the AP@K scores for each race are summed and divided by the total number of races to derive the MAP@K score. This method simplifies the evaluation process by providing a comprehensive measure of prediction accuracy across various events.

3.3.2 Baseline Model

In the absence of a similar project for comparison, the team resorted to a simple baseline model, known as a ”naïve” model. It is also referred to as a walk-forward validation model because it carries forward the results of one result as predictions for the next. Since the research’s goal is to predict the top ten race finishers, the naïve model simply selects the top ten finishers from the previous race as its prediction. Initially, the team considered using the qualifying results of the same race being predicted as the basis for predictions. However, this approach was found to result in lower accuracy to beat (AP@K) compared to the naïve baseline model. Therefore, the team decided against proceeding with it, as a higher score to beat would be a better comparison to demonstrate its ability to make accurate predictions.

The machine learning model must outperform the naïve baseline model. This is crucial to demonstrate the model’s capability to achieve enhancements over a basic version. Failure to surpass the baseline may signal a need for further model refinement or even a reconsideration of the project's viability, as it suggests that the model performs worse than the most straightforward methods.

3.3.3 Prediction Steps

Before comparing models, it is pertinent to outline the prediction process. Predictions are conducted individually for each driver in every race, with each driver being independent of the others. Due to this independence, each driver's dataset differs from others, reflecting the variability in the races they compete in. This discrepancy is evident in the features and lags as they are structured to consider only the information of the most recently completed races of a specific driver. For example, in predicting Carlos Sainz's results for the season's 10th race, after missing races 8 and 9 due to a crash and gearbox issues, we would use data from races 3 to 7. In contrast, for Lewis Hamilton, who raced in 8 and 9, we would look at data from races 5 to 9, showcasing the variation in lagged data between the two drivers.

In simpler terms, the team is essentially developing 20 individual models, one for each driver. Predictions for each driver are made in a rolling manner, meaning that once a prediction is completed for a particular race, the process moves on to the next. The actual result of the previous event is, then, included in the training set for the subsequent prediction.

3.3.4 Model Robustness

Time-Series data have an inherent order which must be maintained. A time-series split was done in model testing to evaluate the model's performance at different folds. The last 15 races (races 1 to 15) of 2023 were considered as part of the test set and after each race prediction, AP@k was evaluated, and the original race data point was added to the training data for each driver. Once all 15 races have been evaluated on their AP@k, the MAP@k for a range of races was computed to serve as the score at varying ranges. To demonstrate, first MAP@k will be the mean of AP@k from races 1-15, followed by the MAP@k of races 2-15, and so on and so forth. This ensures that a model's performance will be evaluated at varying sizes of training and test data. The model's overall score will be the average of the 15 MAP@k scores.

3.3.5 Model Comparison

Apart from the baseline, four regressor models were compared where three out of the four are tree-based ensemble models. Mainly, these four are LinearRegression (LR), GradientBoostingMethod (GBM), XGBoostRegressor (XGB), and RandomForestRegressor (RF).

To assess each model, the average of the MAP@K scores across different time series splits was utilized. This approach ensures that the scores compared reflect both the accuracy and robustness of each model. To illustrate how these scores were derived, predictions were made for each driver in each race using the model. The predicted results for the entire race were aggregated and sorted to determine the predicted rankings. The top 10 predicted drivers were compared with the true top 10 drivers of the race, and the AP@K for the race was computed. Since multiple races were being predicted, the average AP@K scores across all races were calculated, resulting in the MAP@K. Moreover, due to the utilization of varying time series splits or test sets, the scores from different tests were averaged to obtain the final score for comparison.

The team fine-tuned each of the four models to optimize hyperparameters for maximizing MAP@K scores. For tree-based models, various combinations of '*n_estimators*', '*max_depth*', '*max_features*', and '*learning_rate*' were explored. Conversely, for LR, only the base model was utilized without applying any regularization techniques.

4 Results and Discussion

4.1 Model Predictions

Table 4: Machine Learning Model Performance with Hyperparameters

Machine Learning Model	**Hyperparameters**	**Model MAP@k**
Linear Regression	default	64%
Gradient Boosting Regressor	n_estimators: 300 max_depth: 10 learning_rate: 0.1	78%
XGBRegressor	n_estimators: 300 max_depth: 8 learning_rate: 0.05	83%
RandomForestRegressor	n_estimators: 300 max_depth: 1 max_features: 0.3	89%

In evaluating predictions using AP@k, k was set to 10 since the top 10 drivers per race are those who will receive points. To find the best model, evaluation scores were compared to the naïve baseline model which had a MAP@k score to beat of 80%. Linear Regression, which was used to check applicability of linear models, had a score of 64%. Ensemble Tree models had a better score over the Linear Regression Model. Gradient Boosting Regressor scored 78% but was worse compared to the naïve baseline. XGBRegressor and RandomForestRegressor scored better than the baseline at 83% and 89% respectively. The best model that was used is the RandomForestRegressor with hyperparameters of 300 *n_estimators*, *max_depth* at 1, and 30% *max_features* (see Table 4).

The best model was used to predict the outcome of the last four races of the 2023 season (see Appendix B). The AP@k for the last four races are 89%, 100%, 88%, 89% respectively, while MAP@k score in total is 91.5%.

4.2 Explanation Techniques

4.2.1 The Why: Local Interpretable Model-Agnostic (LIME)

LIME is an explanation technique applicable to both regressors and classifiers. It generates explanations by locally learning a simple model (e.g., Linear model) near the prediction. It highlights a user-specified number of the most important features that elucidate why the model made its prediction. Additionally, it provides the actual values of each of these important features. The LIME results for Pierre Gasly's prediction in the Abu Dhabi Grand Prix are found below (Sharma, 2018).

Pierre Gasly's predicted position was above 10th place, or exactly at 10.37. This forecast was driven by the features illustrated in Figure 3. According to LIME, Gasly received this prediction due to the following features: qualifying position, finishing position five races ago, as well as his cumulative constructor points, average lap, and average lap position in the previous race. Emphasizing the top two most significant features, Gasly's prediction outside the top ten was primarily influenced by his strong qualifying performance, placing between 8th and 12th

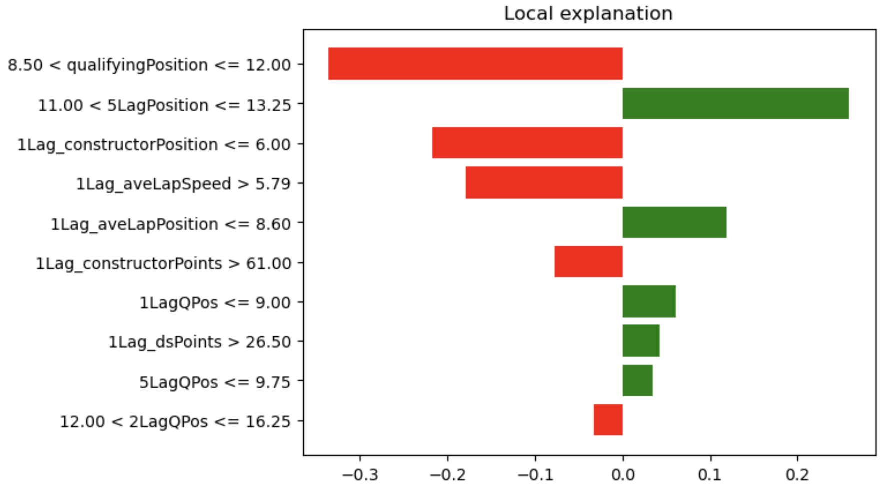

Figure 3: LIME Explanation for Pierre Gasly's Abu Dhabi race in 2023

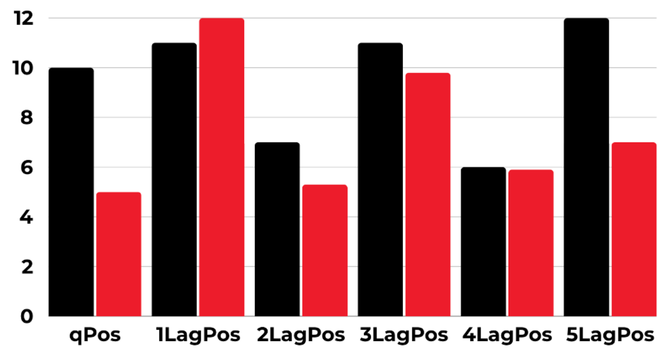

Figure 4: DiCE Counterfactuals for Pierre Gasly's Abu Dhabi race in 2023

place. However, his overall position was negatively impacted by a subpar performance from five races ago, where he finished between 11th and 13th place.

4.2.2 The What if: Diverse Counterfactual Explanations (DICE)

DiCE is an explanation technique that produces counterfactual examples. Counterfactuals represent alterations to the features that would result in a different prediction from the original. DiCE aims to identify the minimal changes to the features necessary to alter the prediction outcome.

Based on the condensed DiCE counterfactuals in Figure 4, the team can identify strategies for Pierre Gasly to enhance his performance and achieve a top-ten prediction five races from now. The primary recommendation is for Gasly to aim for a qualifying result of 5th place in five races' time. Additionally, he should strive for an average finishing position of 8th place across the next five races, as indicated by the adjustments in the 1st to 5th lag positions for race results.

5 Conclusion

Ensemble Tree Models was used to predict the point finishers for Formula One races. F1 data was extracted from Ergast Developer API Website, Wikipedia, and the Official F1 Website. The dataset was constructed by creating time-series features for each driver for each of the races in the 2022 and 2023 seasons. Predictions were done per driver for each race and was evaluated using Average Precision@k at k=10. To test each model's robustness, each model was evaluated using the Mean Average Precision@k at multiple time-series folds and compared with a naïve baseline. The RandomForestRegressor Model (*n_estimators*=300, *max_depth*=1, *max_features*=0.3) outperformed the naïve baseline, as well as other models, with a MAP@k score of 89%. This model predicted the last four races of the 2023 season with the following AP@k scores: 89%, 100%, 88%, and 89% respectively.

LIME was used to understand how predictions were made by the model based on the value of each feature. To illustrate the application of the explanation technique to various drivers, Pierre Gasly was utilized as a case study. It was found that his prediction was largely influenced by his qualifying position, finishing position from five races ago, as well as cumulative constructor points, average lap speed, and average lap position in the previous race. Specifically, his strong qualifying performance was offset by a previous poor race result, contributing to his prediction outside the top ten.

DICE was then used to generate counterfactuals to show how a driver can improve their ranking moving forward. Due to the computational complexity of generating counterfactual examples, the team produced only one for Pierre Gasly to enhance his prediction and be forecasted within the top ten in five races' time. To achieve this, Gasly should aim for a 5th place qualifying result five races from now and strive for an average finishing position of 8th place in his subsequent five races.

Our findings can assist both drivers and constructor teams in comprehending driver performance during races and identifying areas for improvement in rankings. Moreover, the predictions can be valuable for the motor race sports betting and fantasy industry, aiding in the selection of suitable drivers for race events.

6 Recommendations

Considering the limitations encountered during the project, the following suggestions are proposed for future research to improve the performance and accuracy of the model:

- Integrate driver performance metrics into the features for predicting values such as race craft and pace.
- Incorporate car performance metrics, including aerodynamics and engine performance.
- Explore alternative models that allow for batch predictions, such as multi-output neural networks.
- Augment the dataset with real-time race data, such as weather conditions and track temperatures, as well as qualitative insights like team strategy interviews, for more comprehensive analysis.

The team contends that achieving these enhancements would require a partnership with Formula 1 and constructor teams, as they possess the necessary data to fulfill these requirements.

References

[1] Formula 1 in 2022: Explaining the new rules and car changes as teams prepare for first launches.

[2] List of Formula One circuits - Wikipedia, March 2024. [Online; accessed 11. Mar. 2024].

[3] Formula 1® - The Official F1® Website, n.d. [Online; accessed 11. Mar. 2024].

[4] Ergast Developer API. Api documentation, n.d.

[5] Autocom. From racing legends to future stars: Iconic drivers that shaped the formula 1 betting market. 2018.

[6] S. Cooper. 10 big sponsorship deals generating huge $350m fortune on f1 grid. 2023.

[7] V. Efimov. Comprehensive guide to ranking evaluation metrics. 2023.

[8] M. Landry. F1-predictions version 3, 2020.

[9] J. Robert, R. Evrenian, and T. C. Jiyuan Sun. F1 race predictions. Kaggle, 2021, May 26.

Appendix

A Grid Search of Hyperparameters of Ensemble Tree Models

Table 5: GradientBoostingRegressor Model Performance with Hyperparameters

max_depth	**learning_rate**	**MAP@k (%)**
8	0.01	75.33
8	0.05	75.50
8	0.1	76.10
10	0.01	75.17
10	0.05	75.50
10	0.1	76.10
12	0.01	75.17
12	0.05	75.50
12	0.1	76.10

Table 6: XGBRegressor Model Performance with Hyperparameters

max_depth	**learning_rate**	**MAP@k (%)**
8	0.01	81.71
8	0.05	83.16
8	0.1	81.43
10	0.01	80.48
10	0.05	83.16
10	0.1	81.43
12	0.01	80.48
12	0.05	83.16
12	0.1	81.24

Table 7: RandomForestRegressor Model Performance with Hyperparameters

max_depth	**learning_rate**	**MAP@k (%)**
1	0.2	88.82
1	0.25	88.72
1	0.3	89.02
1	0.35	89.02
1	0.4	87.57
2	0.2	88.61
2	0.25	86.86
2	0.3	87.34
2	0.35	87.21
2	0.4	87.34

B Predictions for the Last Four Races of 2023 Season

Table 8: 2023 Mexico Grand Prix Predicted Driver Rankings

Driver	**Predicted Ranking (Sorted)**	**True Ranking**
max_verstappen	2.546	1.0
leclerc	4.020	3.0
sainz	4.206	4.0
hamilton	4.358	2.0
russell	4.902	6.0
piastri	6.898	8.0
norris	7.863	5.0
ocon	8.733	10.0
gasly	10.582	11.0
ricciardo	10.848	7.0
albon	11.613	9.0
zhou	12.874	14.0
bottas	13.446	15.0
tsunoda	14.099	12.0
hulkenberg	14.881	13.0

Table 9: 2023 Las Vegas Grand Prix Predicted Driver Rankings

Driver	**Predicted Ranking (Sorted)**	**True Ranking**
max_verstappen	1.641	1.0
perez	3.992	4.0
hamilton	4.225	8.0
sainz	4.779	6.0
alonso	6.407	3.0
norris	7.404	2.0
ocon	8.700	10.0
piastri	9.146	14.0
stroll	9.635	5.0
gasly	10.327	7.0
ricciardo	11.430	13.0
tsunoda	13.640	9.0
hulkenberg	14.252	12.0
sargeant	14.267	11.0

Table 10: 2023 Brazil Grand Prix Predicted Driver Rankings

Driver	**Predicted Ranking (Sorted)**	**True Ranking**
max_verstappen	1.841	1.0
leclerc	3.946	2.0
perez	4.228	3.0
hamilton	4.238	7.0
sainz	4.553	6.0
russell	5.142	8.0
alonso	6.622	9.0
ocon	8.667	4.0
piastri	9.393	10.0
stroll	9.783	5.0
albon	10.050	12.0
gasly	10.363	11.0
ricciardo	11.117	14.0
bottas	11.272	17.0
sargeant	12.590	16.0
zhou	13.037	15.0
kevin_magnussen	13.842	13.0

Table 11: 2023 Abu Dhabi Grand Prix Predicted Driver Rankings

Driver	**Predicted Ranking (Sorted)**	**True Ranking**
max_verstappen	1.614	1.0
perez	3.836	4.0
leclerc	3.840	2.0
hamilton	4.972	9.0
piastri	5.886	6.0
russell	5.945	3.0
norris	6.368	5.0
alonso	7.469	7.0
ocon	8.443	12.0
stroll	9.076	10.0
gasly	10.374	13.0
albon	11.304	14.0
ricciardo	11.508	11.0
bottas	12.177	19.0
tsunoda	13.492	8.0
zhou	13.565	17.0
hulkenberg	13.756	15.0
sargeant	14.075	16.0
kevin_magnussen	14.168	20.0

C DiCE Counterfactuals

Table 12: Data Feature Changes

Feature	**Original**	**Changes**
1Lag_dsPoints	62.0	-
1Lag_dsPosition	11.0	-
1Lag_driverWins	0.0	-
1Lag_constructorPoints	120.0	-
1Lag_constructorPosition	6.0	-
1Lag_constructorWins	0.0	-
1Lag_aveLapPosition	6.26	-
1Lag_aveLapSpeed	6.201	-
1Lag_fastestLapRank	11.0	-
qualifyingPosition	10.0	5.0
1LagPosition	11.0	12.0
2LagPosition	7.0	5.3
3LagPosition	11.0	9.8
4LagPosition	6.0	5.9
5LagPosition	12.0	7.0
1LagQPos	5.0	-
2LagQPos	13.0	-
3LagQPos	11.0	-
4LagQPos	7.0	-
5LagQPos	7.0	-
positionOrder	10.374	9.536