Feature Importance versus Feature Selection in Predictive Modeling for Formula 1 Race Standings

OCTAVIANA-ALEXANDRA CHETELES, University of Twente, The Netherlands

In the fast-paced world of Formula 1, drivers' skills, technological advancements, and, most importantly, the strategic use of data are all used to gain a competitive edge. This research paper aims to determine how to accurately extract the most influential features of race outcomes that are publicly available. The analysis is conducted using two methods, feature importance and feature selection. One approach explores the division of the features into weather, car, and driver categories to develop specific predictive models, assessing the importance of top features within each category. A final model with each model's set of features that give the lowest root-mean-square is created and compared to the other approach, which is applying feature selection from the beginning to a new model. Additional features are developed based on the existing ones and used in both approaches. To improve prediction model accuracy, the lowest root mean square error (RMSE) possible is targeted, and to evaluate the features, feature importance scores are used. The following set of features was discovered to be the most outcome-significant in all models: the grid position, the average breaking points, and the variance of the breaking points. The importance-based model presented the lowest RMSE, 0.005 and 0.006 when using Random Forest Regression and a Gradient Boosting Regression respectively. The model that used feature selection had a deviation of 0.93 when using Random Forest Regression and 2.23 Gradient Boosting Regression. The RMSE values decreased for all models when new features were added.

Additional Key Words and Phrases: Deep Learning, Formula 1 Racing, Feature Importance, Feature Selection, Predictive Modelling, Machine Learning, Sports Analytics.

1 INTRODUCTION

In Formula 1, success depends not just on technical ability but also on strategic intelligence, which involves interpreting data and making sound decisions. Each race generates a vast amount of data, both in complexity and volume, summing up to more than 1 terabyte [1], which makes them great sources for research and insights that can have a big impact on the results. This will provide teams with data-driven tactics to maximize their performance by exposing the features that are the most important for the final standings. The potential of machine learning to improve racing strategy has also been shown in recent research (e.g. [25]), for predicting the fastest lap time in qualifying.

In the world of racing, success is dictated by a delicate balance between human and machine. Drivers must navigate circuits with unparalleled skill, but their performance is heavily influenced by the capabilities of their cars. Advanced technologies and engineering innovations play a crucial role in car performance, which include aspects such as aerodynamics, engine power, tyre management, and fuel efficiency. Additionally, team strategies around pit stops, tyre choices, and race-day decisions can significantly impact race outcomes.

Racecraft, consistency, and qualifying performance are the main factors that determine a driver's success. Furthermore, the driver-team dynamic is critical since racing tactics, including tyre selection and pit stop scheduling, frequently determine the outcome of close races.

The situation gets even more complex when outside variables including weather, track features, and racing events are taken into consideration. Unfavorable weather may significantly change the dynamics of a race by affecting tyre performance and visibility.

This paper aims to explore the key features that impact the final standings in Formula 1 races the most. By using data-driven analysis and exploring the interactions between various performance indicators, this paper seeks to dive into the complexities of race dynamics. This exploration should provide the teams, engineers, and enthusiasts alike with valuable insights. In order to find underlying trends that influence race outcomes, this research suggests a systematic analysis on past race data. Nevertheless, the method of creating a model based on the best feature-importance combination of others seems quite novel, as no straight-forward research papers were found in this direction. This might be the case as it could take a lot of time computationally and might not be very accurate compared to other techniques. This matter will be studied throughout the paper.

2 RELATED WORKS

This section examines the body of research on feature importance, feature selection methods, feature, and its use in Formula 1 racing and other sports-related predictive modeling.

One metric for determining which variables in a predictive model have the most influence is feature importance. Breiman [4] presented Random Forests, a machine learning technique that determines the relative relevance of features by assessing how each feature affects the correctness of the model. Sports analytics is one of the many fields in which this approach has found widespread adoption. Bunker and Thabtah [22], for example, used machine learning approaches to forecast the results of sporting events, highlighting the significance of features in improving prediction accuracy.

In order to enhance model performance and minimize overfitting, feature selection is used for selecting a subset of useful characteristics. A thorough analysis of feature selection strategies was given by Guyon and Elisseeff [7], who highlighted approaches including filter, wrapper, and embedding strategies. These methods have been applied to sports analytics to improve model performance and simplify datasets. Liu et al. [2], for instance, used feature selection to forecast the results of NBA games, demonstrating how reducing the feature space may result in more effective and precise models.

The prediction ability of models can be improved by extracting additional characteristics from already-existing datasets.

TScIT 41, July 5, 2024, Enschede, The Netherlands

© 2022 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Feature engineering is the practice of developing new variables to identify underlying patterns in the data. Predictive model errors have been demonstrated to decrease mistakes in Formula 1 racing when derived characteristics like tyre degradation rates, pit stop efficiency, and sector-specific performance measures are included. In data science, Katya [10], for instance, studied data scaling, one-hot encoding, and handling missing values, together with feature selection, dimensionality reduction, and interaction term creation. Applying similar strategies to Formula 1 race data can lead to the development of new features, such as tyre wear rates or pit stop efficiency, potentially reducing RMSE and improving the accuracy of race outcome predictions.

Research comparing different feature selection strategies and their effects on model performance in the context of Formula 1 racing is scarce. Studies conducted in similar domains, however, offer insightful information. Shaikhina and Khovanova [19] conducted a comparison of feature selection techniques in biomedical datasets, demonstrating the potential to enhance model accuracy by using feature importance. Similar approaches used for Formula 1 datasets may show how feature importance differs between models and time periods. Moreover, using feature importance to select only the relevant features could improve the accuracy of the models, although it is more time-costly.

3 RESEARCH QUESTIONS

The problem statement leads to the following research question:

How can the publicly available features that have the highest influence on the final standings in a Formula 1 race be efficiently extracted?

This can be answered with the help of the following sub-questions:

RQ1: What new relevant features can be derived from the existing ones in a Formula 1 race dataset?

RQ2: What is the impact of the newly added features on the root mean square error (RMSE) of the predictive models?

RQ3: How does the performance (accuracy-wise) of a predictive model in Formula 1 racing change when using feature importance repetitively, in comparison with feature selection application?

4 METHODS AND APPROACH

For the methodology of this project, the cross-industry process for data mining (CRISP-DM) is used.

4.1 Business Understanding

The goal of the paper is to utilize historical race data to spot trends in performance and decision-making. For this, the Formula 1 databases [12] and APIs, such as the Ergast Developer API [3], which include historical race statistics, driver performance metrics, timing data, weather data, car telemetry, and position data over the last five years (2018–2023), will be used. Prior seasons will not be taken into account as the historical data up until 2018 is not entirely made public by the Formula 1 teams. Moreover, the library FastF1 [18], created by a German engineer [17], uses the data from Ergast to access the previously mentioned data with the addition of custom functions to the Pandas objects to make the data access and work process fast and simple. All data is provided in the form of extended Pandas DataFrames [13], and it can be visualized through Matplotlib [9]. Due to the API being experimental, instability causes data to be occasionally unavailable, resulting in difficulty in extracting data.

4.2 Data Understanding

There are around 22 races per year (season), in which 20 drivers, 2 from the same team, are competing against each other. Each race has different track outlines and weather conditions, which also require certain car adjustments and strategies. An exception to this number of races is the 2020 season, in which only 17 races took place due to the COVID-19 pandemic.

The data is obtained through the following data objects from the FastF1 Python library: 'Session', 'Laps', 'Lap', 'Telemetry', 'SessionResults', each object having object attributes and methods. A 'Session' object contains information about an event; the 'Laps' object specifies information about multiple laps; the laps are looked at one-by-one with the 'Lap' object; 'Telemetry' contains information about the car; and 'SessionResults' provides information mainly about the final standings. The full details can be accessed on the FastF1 library's page [18].

As with any data set, the above-mentioned ones also have irrelevant features, which include identifiers or personal information about the driver or the team, which do not have a direct influence on race outcomes. (e.g.'IsAccurate', 'BroadcastName'). These will be deleted to prevent overfitting, accuracy reduction, and efficiency loss. Moreover, each feature is more or less characteristic of one of the following categories: car, driver, weather, or track.

In this paper, the features will be divided among the first three categories, although in Formula One Racing, the car's construction is the most crucial element for the final standings. Bell et al. [15] analyzed whether driver skill or car construction affected the performance (as shown by points scored) the most using a multilevel (random coefficients) linear model. It was shown that the car's construction accounted for 86% of the variation in points awarded, with driver skill accounting for just 14% of the total. Kesteren and Bergkamp [20] distinguished between constructor advantage and driving competence using a Bayesian Multilevel Beta regression technique and concluded that approximately 88% of the variance in race results is explained by the constructor.

Telemetry in Formula 1 refers to the system of wireless data transmission from the race car to the team's engineers in real time. This technology captures and sends a wide array of data points, including vehicle speed, tyre pressure, engine performance, and more, during a race [5]. Unfortunately, it can have server or upload issues, leaving quite a number of missing or undefined values. Moreover, when looking at the telemetry data, the most relevant car-related features are not publicly available. Those features include tyre pressure, engine temperatures, fuel levels, and brake pressure [5].

4.3 Data Preparation

This analysis is conducted in Python. Because each session provides big data sets and they require a lot of time and memory when reloading, the Pickle Python module [21] and caching are used to speed up the process. Pickle is essentially serializing and deserializing Python objects into or from a file of binary strings. After the data extraction for a model is completed, it is saved in a data frame and then loaded into a Pickle file.

Non-numeric data (e.g., 'Compound', which represents the tyre type, can take the following values: SOFT, MEDIUM, HARD, INTERMEDIATE, WET) will be converted to categorical values, thus a feature for each possible value. This method, suggested in [10], is called hot encoding, and it provides structured information and can improve the performance and accuracy of the model, as the values of the newly-added features are of type boolean (e.g., if a compound is soft, only the 'Compound_SOFT' parameter will take value 1, and the rest, 0).

Data pruning and normalization techniques are applied to handle missing values and outliers and ensure data integrity. In case of a missing value, the numeric data is replaced with an appropriate value, either the mean of the category (lap times, sector times, speeds, etc.) or 0 (tyre life, non-numeric classification position - "R" (retired), "D" (disqualified), "E" (excluded), "W" (withdrawn), "F" (failed to qualify), "N" (not classified), etc.). Time-related features are converted to seconds for easier handling.

For each model, at least the calendar of one season is fetched. The season is specified through a range of years (e.g., 2018–2019), and the model can be created for more seasons by extending the range to more years. The next step is to iterate through each included season's events and update or add new features based on the existing ones. In cases where only a couple of features from a data frame are needed, only the relevant columns are extracted (e.g., the 'SessionResults' data object has two features of interest: 'ClassifiedPosition' (the final position) and 'GridPosition' (the starting position, based on the qualifications).

If two or more data frames are used in an iteration, they are merged together on one or more common features to be able to create a proper data frame for a model. If no common feature exists, they will be merged on the 'Event' feature, which is extracted at every iteration. However, this is not the best approach, as handling and processing merged data becomes more complex.

4.4 Modelling

A purpose of the paper is to analyze the approach that most accurately selects the most important features for standings prediction. Moreover, the models can be analyzed in two ways based on the target, which can either be the final position ('ClassifiedPosition' from 'SessionResults'), which is ideal for predictive modeling, or the lap-by-lap position ('Position'), which provides insights into real-time race strategy and helps with quick decision-making. However, the later one presents a higher risk of overfitting. In this scope, the problem will be tackled in two ways:

Repetitive Application of Feature Importance

For this approach, three separate models are created. The available data is split into three categories: driver performance, weather conditions and telemetry data. A model is created for each category. such that the most important features of each of them will be known. By splitting the features into smaller sets, the strengths of each model become evident. Ultimately, the set of features that create the lowest RMSE is taken from each model and used to create a new one. The final model should only have the most significant characteristics of each category, thus enhancing the prediction accuracy and preventing overfitting with irrelevant features.

Application of Feature Selection

This approach is quite straight-forward. All the newly-thought-out features are added to the existing ones, creating a model with 61 features. The same data preparation steps are applied here. Finally, a list of the most important features is provided, with no category restriction. This model is more complex and, thus, computationally and time-costly. Here, feature selection uses two techniques: *SelectKBest* and *Recursive Feature Elimination (RFE)*. *SelectKBest*, a filter-based feature selection method that relies on statistical measures to score and rank the features, together with the *f_regression* score function, selects the top k features based on the correlation with the target variable [6]. The latter technique improves feature selection progress by recursively removing the least important features. Thus, the model performs better and prevents overfitting by using fewer features. This is needed because this model has all the original relevant values plus the newly created ones, which are the same as in the previous approach.

As mentioned in the previous sections, the target will differ based on the purpose. The features will be assessed on both lap-by-lap and final positioning. As 'Position' is highly correlated with the target variable 'ClassifiedPosition', the last values of 'Position' being identical, this feature will not be taken into consideration. Moreover, as different algorithms have their strengths and weaknesses, the datasets are tested on multiple forest models (Random Forest Regression and Gradient Boosting Machines). Random Forest Regression is used as it is effective in handling large datasets that have many features and missing values or outliers. The Gradient Boosting Tree method. refers to strategically combining additional trees by correcting mistakes made by its previous base models. Hence, it potentially improves prediction accuracy, as it can also be seen in travel time prediction [24]. The models for both algorithms are computed with 100 decision trees and a random number generator seed of 42.

4.5 Evaluation

The evaluation of the models' performance will be assessed through the Root Mean Squared Error (RMSE). This is advised in [8], as the errors are Gaussian (measured with the same instrument under seasonal constant conditions). Regression projects often use the RMSE statistic to determine the average size of the mistake. The square root of the average squared discrepancies between the expected and actual values is used to compute it. Based on its value, new features will be added or removed, such that the RMSE becomes as low as possible.

The features will be evaluated by the in-built feature importance list. SHAP values were also taken into consideration. However, they can be computationally expensive and time-consuming for large data sets. In [23], it is also stated that opting for the model's built-in feature importance list can offer a more efficient and practical approach for larger datasets and more intricate models.

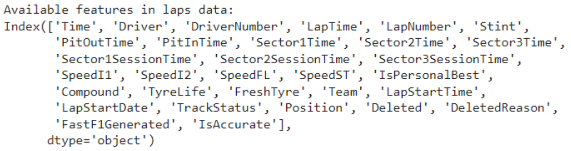
4.6 Deployment

The goal of the deployment phase is to use the built-in predictive models to extract useful information from Formula 1 race data. To enable smooth data processing and retrieval, this entails integrating the models into the current data pipeline. Additionally, a retrospective about what went well, what could have been better, and how to improve in the future will be included.

5 RESULTS

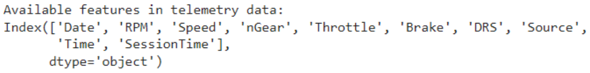
For this analysis, the 2018 season was arbitrarily picked.

All the available features are listed in the figures below (*Figure* 1, *Figure* 2, *Figure* 3 and *Figure* 4).



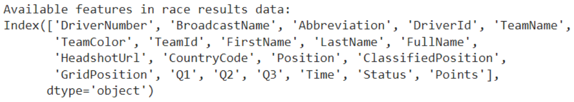
```
Available features in laps data:
Index(['Time', 'Driver', 'DriverNumber', 'LapTime', 'LapNumber', 'Stint',
       'PitOutTime', 'PitInTime', 'Sector1Time', 'Sector2Time', 'Sector3Time',
       'Sector1SessionTime', 'Sector2SessionTime', 'Sector3SessionTime',
       'SpeedI1', 'SpeedI2', 'SpeedFL', 'SpeedST', 'IsPersonalBest',
       'Compound', 'TyreLife', 'FreshTyre', 'Team', 'LapStartTime',
       'LapStartDate', 'TrackStatus', 'Position', 'Deleted', 'DeletedReason',
       'FastF1Generated', 'IsAccurate'],
      dtype='object')
```

Fig. 1. The publicly available features in laps data



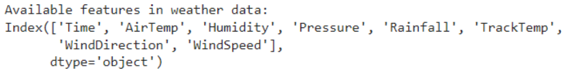
```
Available features in telemetry data:
Index(['Date', 'RPM', 'Speed', 'nGear', 'Throttle', 'Brake', 'DRS', 'Source',
       'Time', 'SessionTime'],
      dtype='object')
```

Fig. 2. The publicly available features in telemetry data



```
Available features in race results data:
Index(['DriverNumber', 'BroadcastName', 'Abbreviation', 'DriverId', 'TeamName',
       'TeamColor', 'TeamId', 'FirstName', 'LastName', 'FullName',
       'HeadshotUrl', 'CountryCode', 'Position', 'ClassifiedPosition',
       'GridPosition', 'Q1', 'Q2', 'Q3', 'Time', 'Status', 'Points'],
      dtype='object')
```

Fig. 3. The publicly available features in race results data



```
Available features in weather data:
Index(['Time', 'AirTemp', 'Humidity', 'Pressure', 'Rainfall', 'TrackTemp',
       'WindDirection', 'WindSpeed'],
      dtype='object')
```

Fig. 4. The publicly available features in weather data

Separating the contribution of each feature was made difficult due to the complex interaction between the driver and the car, as also stated in [16]. After careful consideration, according to the relations mentioned in **Data Preparation** and the actual significance of the features (see [18]), the feature division and selection of relevant features can be found in the list below:

- **Driver Model**: Driver Number, Times of the Sectors (1, 2 and 3), Times of the Sessions of the Sectors (1, 2 and 3), Lap Times, Position, SpeedFL, SpeedST, Speed1, Speed2, Stint, Throttle, Grid Position, Classified Position
- **Car Model**: Team, Revolutions per Minute (RPM), Speed, Stint, Compound, Tyre Life, Freshness of the Tyre
- **Weather Model**: Air Temperature, Humidity, Pressure, Rainfall, Track Temperature, Wind Direction, Wind Speed

The only strictly car and team-related features are 'RPM' and 'Speed' from the telemetry data, and 'Stint', 'Compound', 'TyreLife', 'FreshTyre' from laps data. The throttle and braking characteristics were included in the additional features, as they do not provide any useful insight in the initial raw form. The pit stop times and other possible related features are highly team-dependent as well. However, as they could not be extracted correctly or are mostly missing, they did not represent noise-free data. The weather model would use all of its features, except for 'Time'. Lastly, the driver model had all the other significant features.

By creating a model for each category only using the enlisted features, the RMSEs looked as in Table 1, where 'P' represents the 'Position' as target, and 'C', the 'ClassifiedPosition'.

	Random Forest	Gradient Boosting Regressor
Driver P	1.28	2.34
Driver C	1.53	2.81
Weather P	5.21	5.21
Weather C	5.08	5.08
Car P	2.2	2.6
Car C	1.95	2.35

Table 1. RMSE values of the original features

In order to reduce the deviation of each model, new features based on the existent ones were added. For the driver model, the following features were included:

- *Cumulative lap time* (the sum of all lap times) - It gives insight into the trend of a driver. If it constantly increases, it may indicate a decline in performance, which is usually because of tyre wear or driver fatigue.
- *Lap time variance* - It measures the consistency of a driver's lap times over a series of laps. The higher the variance, the greater the inconsistency. In correlation to consistent lap times comes the driver's skill, car setup and stable track conditions.
- *Fastest lap time* - It provides a benchmark for a driver's performance.
- *The number of braking points* - Speed and throttle position are analyzed to identify how and where a driver decelerates. This feature is directly dependent on the driver's skill, being the only one who chooses when to break, which is crucial in context. A higher number of braking points can indicate a

more aggressive driving style, which can lead to higher tyre wear, which might necessitate a pit stop for a tyre change, or, in extreme cases, a crash. Additionally, the mean and the variance for the breaking points will be analyzed.

- *Gear shift smoothness* - Gear shifts are handled by the driver solely as well. Based on them, the gear shift smoothness, or more explicitly, the variability in gear changes, the average number of gear shifts per lap, and the gear shift variance can be computed. The mean and variance will be determined for this feature as well.
- The averages of all speed- and time-related features from the driver model are shown in the afore-mentioned list. They show the general performance trends, which are the interest points, rather than inconsistencies in performance.

When analyzing the difference between the feature standings from the original model (*Figure* 5) and the additional-features model (*Figure* 6), it could be seen that the first position was still occupied by 'GridPosition', while the following features were taken by some feature related to drivers skill. Some examples are the variance of the braking points and the average of gear shifts. As the fastest lap represents the best time obtained by a driver during a race while doing a lap, it was only logical to have a big impact on the final standings and take second place as the most important feature in the reevaluated driver model. In these plots, RFR stands for Random Forest Regression. The plots for the Gradient Boosting Regression are listed in the appendix (*Figure* 15 and *Figure* 16).

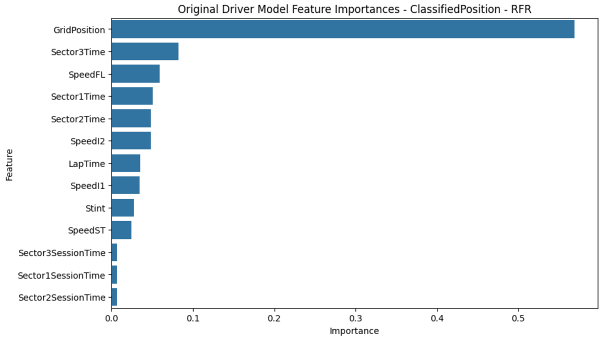


Fig. 5. Driver Model Original Feature Importances - RFR

For the weather model, for each of the eight characteristics, the mean and standard deviation, together with the rolling mean and variance for a window of 5 were calculated. The maximum and minimum values, plus the change between the values from the start of the session and the end of the session were added as well. Additionally, the interaction between the air temperature-humidity and wind speed-wind direction pairs was studied.

For the remaining telemetry features (engine revolutions per minute (rpm) and speed), the mean and the variance were additionally determined.

After adding the above-mentioned features and then playing with the features to see which combination reduces the root-mean-square deviation the most, some values dropped to half of the original ones.

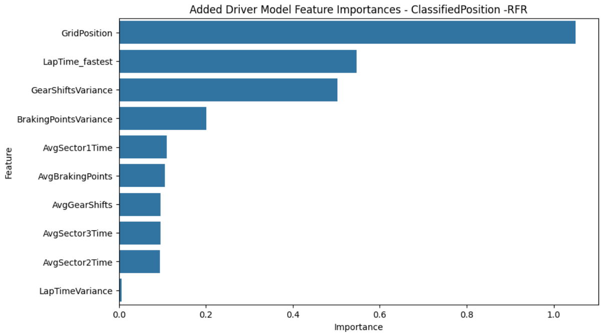


Fig. 6. Driver Model Added Feature Importances - RFR

When looking at the weather data frame, it was quite hard to connect it to the final standings, as there were no common features. For this purpose, the connection has been made to the new 'Event' feature. For the initial six values, the following order has been seen when looking at lap-by-lap positioning: air temperature, humidity, pressure, track temperature, wind direction, wind speed. Curiously, when the target value changes, the first three interchange positions, with humidity going on in the first place, followed by pressure and air temperature. As rain is generally highly correlated with high pressure before it starts and high humidity before and during it, this could be considered a highly reasonable order, as also suggested by the Red Bull team [11]. The plots for both algorithms can be seen in *Figure* 7 and *Figure* 8. The other plots of the algorithms on the original-features and additional-features plots can be seen in the appendix (*Figure* 17 and *Figure* 18). However, the weather deviation slightly changed when new features were added. Therefore, a set of the three most target-relevant features was extracted.

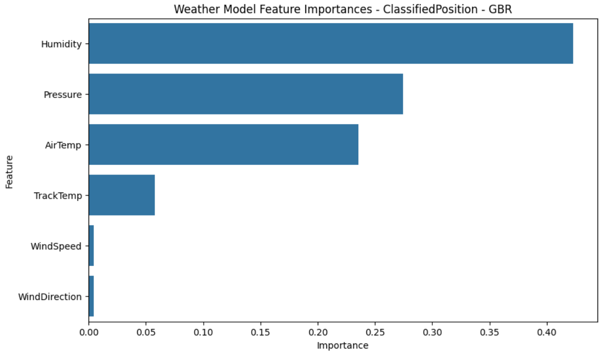


Fig. 7. Weather Model Original Feature Importances - GBR

When using hot-encoding for the team names in the car model, the RMSE decreased significantly, dropping from 3.56 to 1.72. The according plot can be seen in *Figure* 9. In comparison to the initial car model, when hot-encoding was used for team names and tyre compounds in the model with the additional features, the RMSE increased from 1.75 to 3.26. However, when it was applied only to the 'Compound' feature, the RMSE decreased again to 1.62. As there

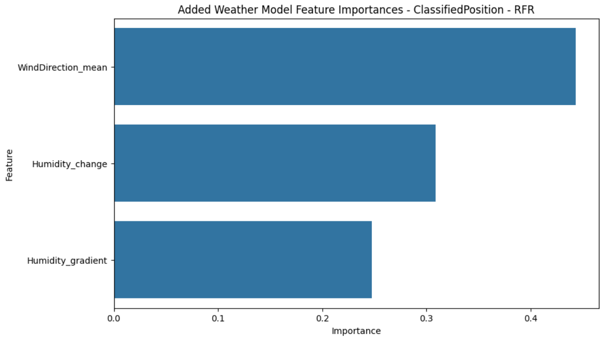


Fig. 8. Weather Model Added Feature Importances - RFR

were only five types of tyres, but ten teams, the complexity could have risen too much with the additional encoding of the team names. Moreover, the importance of the top features changed according to *Figure* 10.

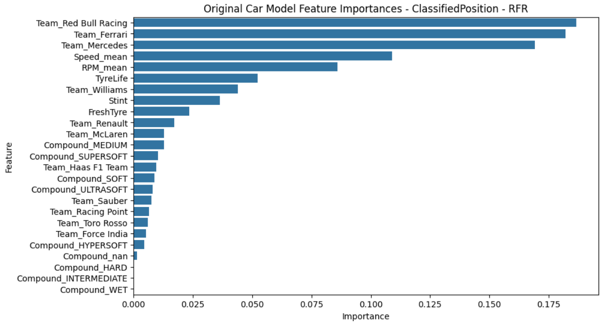


Fig. 9. Original Car Model Feature Importances with Team and Compound Encoding

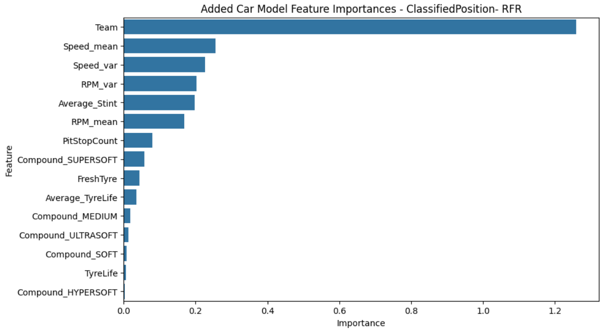


Fig. 10. Car Model Added Feature Importances - RFR

After applying the adjustments, the final values for RMSE changed according to Table 2.

Below, the importance-ordered feature sets that determined the above-mentioned RMSE values can be seen:

Table 2. RMSE values with the added features

	Random Forest	Gradient Boosting Regressor
Driver P	0.81	2.14
Driver C	0.24	2.10
Weather P	5.08	5.20
Weather C	5.08	5.08
Car P	1.62	3.82
Car C	1.91	1.91

- Driver model: { GridPosition, LapTime_fastest, BrakingPointsVariance, Avg-GearShifts, AvgSector3Time, GearShiftsVariance, AvgSector2-Time, AvgSector1Time, LapTimeVariance, AvgBrakingPoints }
- Weather model: { Humidity_gradient, Humidity_change, Humidity_mean }
- Car model: { Team, Speed_mean, RPM_var, Speed_var, Average_Stint, Average_TyreLife, RPM_mean, FreshTyre, PitStopCount, TyreLife, Compound_SOFT, Compound_SUPERSOFT, Compound_MEDIUM, Compound_ULTRASOFT, Compound_HYPERSOFT }

When the feature sets mentioned above were combined for the final model and had 'ClassifiedPosition' as the target, they resulted in a root-mean-squared deviation of 0.005 for the Random Forest Regression model and 0.006 for the Gradient Boosting Machine model. This indicates that the average difference between the predicted values and the actual values is very small. As it can be seen in *Figure* 11, the most important features for the final race standings are the starting position ('GridPosition') and the fastest lap time obtained by the driver. Interestingly, the following two places were taken by certain weather features, and only then the driver's skill-related features followed ('BrakingPointsVariance', 'GearShiftsVariance', 'AvgGearShifts', 'AvgBrakingPoints'). On the Gradient Boosting Machine, the order was not the same, as it can be seen in *Figure* 12.

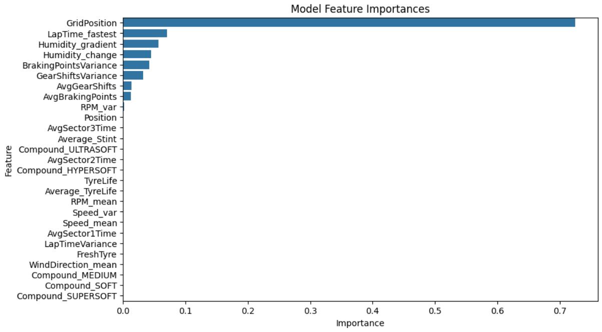


Fig. 11. Combined Model Feature Importances - RFR

The values of the root-mean-square deviation differed when making the lap-by-lap analysis, the values being 0.77 and 2.26 for the

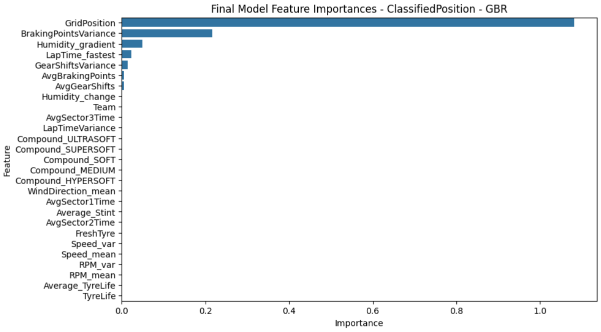


Fig. 12. Combined Model Feature Importances - GBR

Random Forest Regression and Gradient Boosting Machine, respectively. The values with regard to real-time analysis were higher. However, this paper focused on the model prediction parameter as target value ('ClassifiedPosition'), computing the other one purely for comparison and future work possibilities.

For the feature selection approach, all the previously mentioned new features were added to a data frame, together with the existing ones. The total number of features rose to 112. The same data preparation was performed as in the other case. Compared to the other model, the feature list differed in order and RMSE. The lowest deviation was obtained on the Random Forest Regressor Tree, with a value of 0.93, whereas on the other model, the RMSE value was 2.23. The top-most important features are shown in *Figure* 13, modeled on the Random Forest Regressor, and in *Figure* 14, modeled on the Gradient Boosting Machine.

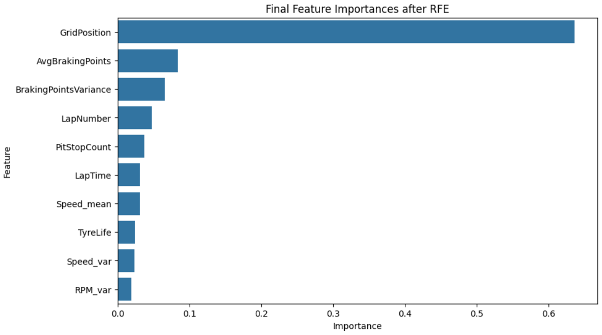


Fig. 13. Final Feature Importances - Selection - RFR

6 DISCUSSION AND CONCLUSIONS

Ankur Patit et al. [14] conducted a study on the correlation between the features and came to the conclusion that variables such as average pole position, number of laps led, and tyre types show significant correlations with race results and driver performance. The following are the key insights from the correlation plot in their paper:

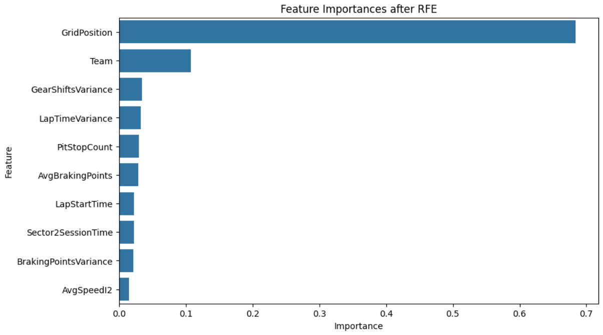


Fig. 14. Final Feature Importances - Selection - GBR

- The average pit stop shows a moderately positive correlation with the use of medium tyres and a negative correlation with the use of super-soft and ultra-soft tyres.
- The tyres have seven types (hard, medium, soft, super soft, ultra soft, wet, and intermediate). Soft and super-soft tyres have a slight positive correlation with the first, second, and third positions. Ultra-soft tyres are negatively correlated with higher finishing positions, as they are more prone to causing accidents.

As it can be noticed, the previously obtained driver feature set is supported by Ankur's findings.

Based on the points reached in the paper, the best performance was achieved by applying the first strategy, which is dividing the large data frame into smaller data frames. Using feature importance in combination with feature importance permutation, the best feature-set was formed by the characteristics that provided the lowest root-mean-square deviation from all categories. Feature importance was used once again to determine the right order of the ultimate set of features. The most significant features for the final standings were the grid position, the average breaking points, and the variance of the breaking points with a root-mean-square deviation of 0.005. This result was obtained using the Random Forest Regressor, while the other algorithm provided a deviation of 0.006 for the same set of features. The low root-mean-square deviation shows that the difference between the predictions and the actual values is extremely low, thus the accuracy of the model scores high.

However, the grid position, the average breaking points, and the variance of the breaking points appeared in all final models, making them the most relevant features when it comes to predicting the final standings.

Although the values should have been smaller for the Gradient Boosting Machine, they seemed to be greater than the ones given by the other algorithm. It might be that it takes longer to work with the Boosting algorithm, as the trees are trained sequentially in comparison with the parallel training from the Random Forest algorithm.

The Jupyter notebooks, together with extra plots, including the real-time telemetry plots concerning the development of the models, are available through a public GitLab repository. [1].

[1] https://gitlab.com/o.cheteles /feature-importance-project.git

7 FUTURE WORK

As future work, it would be wise to ensure a system like Ergast that can be hosted locally so that this project does not rely on an external server, especially as the Ergast Motor Racing Database API will be shutting down at the end of 2024. Moreover, dimensionality reduction techniques could be used instead of feature importance, resulting in faster computational speed and higher model accuracy while preserving more information about feature relationships. Moreover, the change in the feature order over the years could be analyzed, as it could also reflect the changes commanded by the Fédération Internationale de l'Automobile, the governing body of motor sport, and promote safe, sustainable, and accessible mobility for all road users across the world.

ACKNOWLEDGMENTS

The author would like to thank Faizan Ahmed for his guidance throughout this research and Bianca-Maria Filip for the inspiration on the subject.

During the preparation of this work, the author used ChatGPT to structure this LateX document and correct the grammatical errors in it. After using this tool or service, the author reviewed and edited the content as needed and took full responsibility for the content of the work.

A APPENDIX

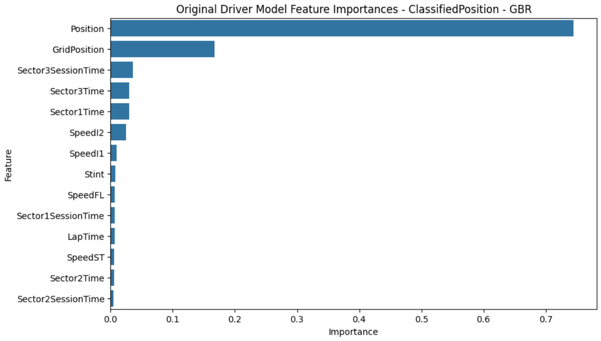


Fig. 15. Driver Model Original Feature Importances - GBR

REFERENCES

[1] Mercedes-Benz AG. 2024. Feature: Data and Electronics in F1, Explained! https://www.mercedesamgf1.com/news/feature-data-and-electronics-in-f1-explained

[2] Roger Alonso and Marina Bagic Babac. 2022. Machine learning approach to predicting a basketball game outcome. *International Journal of Data Science* 7, 1 (01 2022), 60. https://doi.org/10.1504/IJDS.2022.124356

[3] Ergast Developer API. [n. d.]. Ergast Developer API. http://ergast.com/mrd/. Accessed: 2024-04-23.

[4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[5] Catapult. 2024. F1 Technology Data Analysis: How data is transforming race performance. https://www.catapult.com/blog/f1-data-analysis-transforming-performance

[6] Kavya D. 2023. Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning. https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48

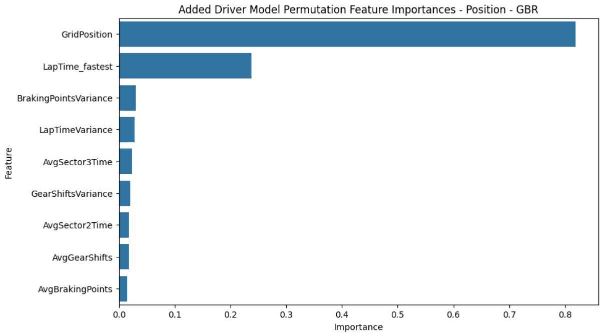


Fig. 16. Driver Model Added Feature Importances - GBR

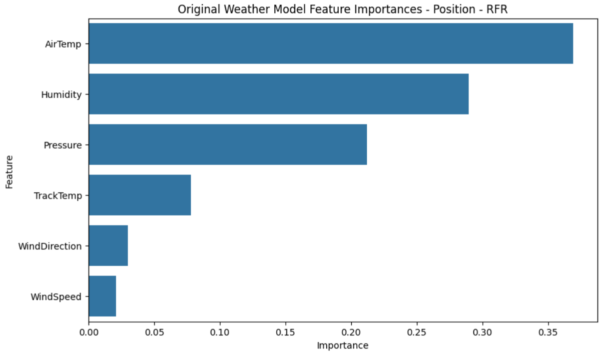


Fig. 17. Weather Model Original Feature Importances - RFR

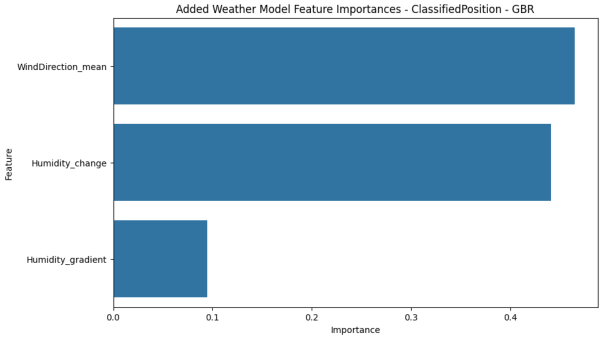


Fig. 18. Weather Model Added Feature Importances - GBR

[7] Isabelle Guyon and André Elisseeff. 2003. An Introduction of Variable and Feature Selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection* 3 (7-9) (01 2003), 1157 – 1182. https://doi.org/10.1162/153244303322753616

[8] T. O. Hodson. 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development* 15, 14 (2022), 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

[9] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[10] Ekaterina Katya. 2023. Exploring Feature Engineering Strategies for Improving Predictive Models in Data Science. *Research Journal of Computer Systems and Engineering* 4 (12 2023), 201–215. https://doi.org/10.52710/rjcse.88

[11] Paul Keith. 2024. How does the weather impact F1 teams and drivers? https://www.redbull.com/int-en/how-weather-impacts-f1-racing#5-humidity

[12] Frank Van Laere. 2024. Formula 1 database | Pitwall. https://pitwall.app/

[13] The pandas development team. 2020. *pandas-dev/pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134

[14] Ankur Patil, Nishtha Jain, Rahul Agrahari, Murhaf Hossari, Fabrizio Orlandi, and Soumyabrata Dev. 2023. A Data-Driven Analysis of Formula 1 Car Races Outcome. In *Artificial Intelligence and Cognitive Science*, Luca Longo and Ruairi O'Reilly (Eds.). Springer Nature Switzerland, Cham, 134–146. https://doi.org/10.1007/978-3-031-26438-2_11

[15] Duane W. Rockerbie and Stephen T. Easton. 2022. Race to the podium: separating and conjoining the car and driver in F1 racing. *Applied Economics* 54, 54 (2022), 6272–6285. https://doi.org/10.1080/00036846.2022.2083068

[16] Duane W. Rockerbie and Stephen T. Easton. 2022. Race to the podium: separating and conjoining the car and driver in F1 racing. *Applied Economics* 54, 54 (2022), 6272–6285. https://doi.org/10.1080/00036846.2022.2083068

[17] Philipp Schaefer. 2021. Python package for accessing and analyzing Formula 1 results, schedules, timing data and telemetry. https://github.com/theOehrly/Fast-F1

[18] Philipp Schaefer. 2024. Python package for accessing and analyzing Formula 1 results, schedules, timing data and telemetry. https://docs.fastf1.dev/

[19] Torgyn Shaikhina and Natalia A. Khovanova. 2017. Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial Intelligence in Medicine* 75 (2017), 51–63. https://doi.org/10.1016/j.artmed.2016.12.003

[20] 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 (July 2023), 273–293. https://doi.org/10.1515/jqas-2022-0021

[21] Guido Van Rossum. 2020. *The Python Library Reference, release 3.8.2*. Python Software Foundation.

[22] William Villegas-Ch, Joselin García-Ortiz, and Angel Jaramillo-Alcazar. 2023. An Approach Based on Recurrent Neural Networks and Interactive Visualization to Improve Explainability in AI Systems. *Big Data and Cognitive Computing* 7, 3 (2023). https://doi.org/10.3390/bdcc7030136

[23] Liang-Q. Hancock J.T. et al. Wang, H. 2024. Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods. *Journal of Big Data* 11 (03 2024). https://doi.org/10.54254/2755-2721/47/20241191

[24] Yanru Zhang and Ali Haghani. 2015. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies* 58 (2015), 308–324. https://doi.org/10.1016/j.trc.2015.02.019 Big Data in Transportation and Traffic Engineering.

[25] Zhixuan Zhao. 2024. Deep Neural Network-based lap time forecasting of Formula 1 Racing. *Applied and Computational Engineering* 47 (03 2024), 61–66. https://doi.org/10.54254/2755-2721/47/20241191