# Introduction with Value Based Problem Statement

In this section, the aim is to pique the reader's interest, underline the relevance of the issue, and pave the way for discussions.

## Introduction:

This introduction is crucial to the project since it determines whether a driver will finish in the top three in a forthcoming race. With its thrilling performances and fast-paced races, the world of racing has a lot of attraction. As a fan of F1 races, I always sought after the top 3 finishes and often compete with others on the prediction of the podium. The difficulty, though, lies in correctly forecasting which drivers will get these top places.

## The Value Based Problem Statement:

The Value Based Problem Statement presented here focuses on our primary objective: developing a model capable of forecasting a driver's chances of placing on the podium in an upcoming race. The significance of this endeavour is felt by everyone involved in the racing industry, including teams, bettors, commentators, and fans. An accurate predictive model might have far-reaching effects, possibly changing race tactics and improving betting forecasts.

## Complexity of our dataset:

Our approaches to data wrangling and transformation, as well as the machine learning methods used for predicting insights, will all be covered in more detail in the next sections. The core of our study goes beyond the binary values of race results, exploring the complexities of predictive modelling for an F1 race.

## Conclusion:

In conclusion, the Introduction with Value-Based Problem Statement acts as a compass, directing readers to the significance and relevance of my effort. In this report, we will emphasize the practical implications of our pursuit by presenting the challenge of predicting podium finishes. This part establishes a connection between the fascination of F1 races and the investigation of data-driven insights.

# Problem Formulation

This study sets out on a quest to explore the complexities of this value-based task, demonstrating the processes taken to load and analyse the data and, in the end, to create a prediction problem that captures the essence of F1 competitiveness.

## Loading and Exploring the Data:

The collection and loading of various datasets that collectively define the environment of F1 competitions is the first step in addressing this challenge. The prediction model's insights are made up of race history, driver profiles, team performance, course features, and season conditions. These various components can be combined to create a thorough perspective of the racing system, which serves as a basis for building the prediction problem.

## Understanding the Data:

To uncover significant trends and insights that influence podium results, a thorough analysis of the data is required. Each dataset tells a narrative; factors such as the track record of the driver, the constructor standings, the location of the track, and the seasons all have a significant impact on the result. The dynamics of the data are shown via visualizations, summary statistics, and exploratory analysis, which enables the discovery of trends that may help formulate the prediction problem.

## Formulating a Prediction Problem:

The development of a prediction problem aims to capture the crux of F1 competitiveness. The challenge is developing a model that can forecast whether a driver will place among the top three in a forthcoming race. With that, I hope to understand the complex interactions of factors that affect podium success. The key to solving the prediction problem is to build features that encompass a driver's previous performance, drivers’ particulars, drivers’ starting position, and other variables that affect race results.

## Approaches and Considerations:

The complexity of racing competition emerges as the problem formulation process progresses. Strategic choices, weather variations, technological difficulties, and driver psychology, adds levels of unpredictability to the assignment. In order to create a problem that captures the spirit of racing while allowing the predictive model to identify the underlying patterns, the formulation process must take these factors into account.

## Conclusion:

In conclusion, the process of formulating the problem, investigating the data, and developing the prediction problem demonstrates the complexity of predicting podium positions in F1 race. The importance of knowing the details of the data and converting them into a predictive framework has been reinforced in this report. The report lays the foundation for the subsequent phases, when data wrangling, cleaning, and model development takes place, as the assignment progresses. The complex methods that led to the creation of the predictive model will shed light on the top three finishers in a setting where milliseconds and careful moves can be the difference between winning and losing.

# Data Wrangling on multiple tables

Predictive modelling for podium finishes in F1 races requires pulling features from many tables and merging them in a way that makes sense. This section goes into the specifics of these actions, emphasizing the processes necessary to build a reliable and thorough dataset that serves as the basis for the model.

## Extracting Features from Different Tables:

In the first step of this project, important details are extracted from many tables that jointly cover the many facets of motorsport competitions. Race results, driver standings, race specifics, circuits, and driver profiles are loaded into this model. The groundwork has been laid for the development of features that capture the dynamics determining podium finishes by utilizing this wide variety of data.

## Table Integration through Concatenation and Merging:

Integrating tables is essential to combining the several dimensions into a single dataset. The core of this integration process consists of merging and joining datasets. The individual tables are linked together using identifiers like raceId, driverId, and circuitId. The resulting dataset provides a comprehensive representation of F1 insights.

### Results and Driver Standings:

Using raceId and driverId as common keys, the results and driver\_standings databases are combined. By integrating race results and driver performance, this process produces a consistent picture of each driver's performance across different races.

### Merging with Races and Circuits:

Using raceId as the common identifier, the merged data is further linked with the races table. This stage expands the dataset by adding race-specific features. The integration then extends to the circuits table, where the circuit information for each race is combined using the circuitId as a point of reference.

### Merging with Drivers:

The dataset and the drivers table are combined in the last stage of integration. This stage adds driver attributes using the driverId as a reference, completing the data required for the predictive analysis.

## Data Transformation and Feature Creation:

The dataset must be refined for predictive modelling when the tables are smoothly merged. This requires data transformation and feature creation. Unneeded columns that have no direct influence on the prediction issue are found and then deleted. This curation process makes sure the dataset stays condensed and relevant, ready to produce precise insights.

The process of feature development is crucial to predictive modelling. Therefore, the binary podium\_finish feature is created to show whether or not a driver finished on the podium (positions 1 to 3), using the position\_x column, which shows the finishing position of a driver in a race.

## Conclusion:

In conclusion, the foundation of predictive modelling for podium finishes in racing is the process of extracting information from many tables and merging them. This section has highlighted the actions required in balancing the multiple components of racing, illuminating the process from data extraction through feature creation and table integration. As the research moves forward, the integrated dataset is ready to make precise predictions. The following sections will expand on this integrated dataset and demonstrate how predictive modelling can dramatically alter the fascinating world of F1 races.

# Data Cleansing and Transformation

In this section, we examine how missing values are handled at various stages, how outliers are handled, how categorical and numerical data are transformed, and how subtle considerations are made to guarantee the dataset's integrity and applicability.

## Missing Value Imputation:

Correcting missing values is essential to ensuring the accuracy of the dataset. We employ methods of strategic imputation to the following features in this context:

### wins:

Imputed with 0 to distinguish between real wins and missing values. This is consistent with the notion that a missing value suggests that the driver did not win.

### fastestLap and fastestLapTime:

Imputed with -1 as an arbitrary value to differentiate between missing values and situations where a driver failed to set a fastest lap or the data is not available.

### date:

In order to maintain chronological accuracy, missing dates are imputed based on raceID by taking use of the ordered nature of the dates with regard to "raceId."

## Others:

The creation of feature "personalBest" is a crucial transformative step. Since the variables 'fastestLap' and 'fastestLapTime\_ms' were unknown before to the race, they were swapped out with 'personalBest'. The fastestLapTime\_ms value with the lowest value, which is obtained by grouping the data by "driverId," is used to calculate this variable.

## Outlier Handling:

To prevent skewed analyses, managing outliers is essential. Considerations include the following:

### wins:

After thorough testing, it is clear that outlier control techniques impair distribution, thus the initial values are kept.

### laps:

Like 'wins', outlier handling techniques have a negative impact on distribution, which results in the retention of initial values.

## Numerical Transformation:

To guarantee that they satisfy the modelling requirements, numerical features are transformed:

### 'podium\_finish':

Since it is already a binary variable (either 0 or 1), no change is required.

### 'raceId' and 'driverId':

These categorical identifiers are left alone because their original qualities are essential.

### grid and wins:

Despite experimenting with other approaches, it is decided to keep the values as they were in order to protect their fundamental value.

### year:

It has been determined that no transformations are necessary after experimenting with it.

### laps:

The middle distribution is elevated, and the extreme tails are suppressed using the Yeo-Johnson transformation.

### dob\_year:

Reciprocal transformation is used to improve the skewness and distribution.

### personalBest:

Power transformation is chosen to reduce skewness and improve distribution.

## Categorical Transformation:

### 'Country' and 'Nationality':

These nominal categorical variables are transformed using one-hot encoding.

## Binning and discretization:

Discretization is important when working with numerical data:

### 'laps', 'grid', and 'personalBest':

Retaining original values owing to unsuccessful results of transformation techniques, maintaining their original value.

### 'wins':

The use of Equal Frequency discretization, which groups data points into bins with constant frequencies, enables outlier management.

### 'year' and 'dob\_year':

Equal Width discretization is used to group data into discrete time periods with consistent bin widths, facilitating analysis within certain time periods.

## Conclusion:

In conclusion, this section highlights the process used in data transforming and cleansing for precise predictive modelling. The dataset's integrity is maintained by dealing with missing values, handling outliers, and transforming categorical and numerical data. As a result, the model is better prepared for further study.

# Machine Learning Model

This section explains how the machine learning model is created and tested against a naive baseline to build an accurate predictive model. The report provides information about the model's performance by showing the counts of rows and columns and assessing the model's effectiveness using classification metrics.

## Feature Importance Analysis:

The dataset's features are examined first. 'df\_logreg\_ss\_feature\_importance' is a structured DataFrame that contains feature labels and the logistic regression model-derived importance scores that correspond to each label. Finding low-importance characteristics with the use of this analysis allows for their removal from the dataset. The adjusted X train and test datasets have 17441 rows and 19 columns as their final dimensions.

## Naïve Baseline Evaluation:

A simple baseline model is created before creating the machine learning model. The baseline accuracy is calculated to be 0.87, and it serves as a standard by which the performance of the succeeding model can be measured.

## Random Forest Classifier:

Random Forest Classifier is used to build the predictive model. The X train and y train datasets were used to train this classifier, which has a maximum depth of 7 and a random state of 42. Based on the X test data, predictions are then produced.

## Model Evaluation:

Important categorization metrics are carefully used to assess the model's performance.

### Accuracy:

The accuracy score evaluates the percentage of instances in the test dataset that were properly predicted. The model has an accuracy of 0.8985, which means that 89.85% of the predicted results were achieved.

### Precision:

Precision measures the proportion of accurate positive forecasts to all positive predictions. A precision score of 0.7435 indicates that 74.35% of the model's optimistic predictions are in fact correct.

### Recall:

Recall measures the proportion of true positive forecasts to all real positives. It is sometimes referred to as sensitivity or true positive rate. According to the recall score of 0.3031, the model is believed to have captured 30.31% of genuine positive cases.

### F1-Score:

The harmonic mean of recall and precision is known as the F1-score. The model balances precision and recall with a value of 0.4306, showing a reasonable trade-off between accurate positive prediction and overall positive instance capture.

## Conclusion:

In conclusion, the machine learning model predicts podium finishes for drivers in upcoming races with a level of accuracy and precision. The evaluation measures shed light on the model's advantages and shortcomings. By outperforming the naive baseline, the model demonstrates its capacity to identify important trends in the data. The thorough examination in this part is a crucial first step in understanding the performances in the F1 races.

# Summary and Further Improvements

Predicting podium results in Formula 1 races is the objective, and the Introduction, Data Wrangling, Data Cleansing and Transformation, and Machine Learning Model sections has given a thorough understanding of it. Each part has offered significant insights that will serve as the cornerstone of the data-driven approach to this problem.

The introduction introduced the value-based problem statement in addition to pique readers interest in the realm of racing. For many stakeholders, such as teams, bookmakers, commentators, and fans, the importance of correctly predicting podium results was stressed. This demonstrates the applicability of our research to the world of F1.

The thorough extraction and integration of several datasets including race results, driver standings, circuit information, and driver profiles were on display during data wrangling, the first practical stage. A complete dataset that serves as the foundation for predictive modelling was produced by this integration. It is essential to have a solid comprehension of the data in order to identify trends and insights that eventually affect podium results.

The dataset's dependability was greatly enhanced by the data cleansing and transformation procedure. The treatment of missing values, outlier control, transformation of numerical and categorical features, and discretization were all covered in detail. These procedures assisted in cleaning up the dataset so that the machine learning model could use it.

The development of the machine learning model, in which we used the Random Forest Classifier to forecast podium positions, was a highlight. The model's capabilities were assessed by measuring its performance using metrics including accuracy, precision, recall, and F1-score. The model's ability to find patterns in the data was demonstrated by outperforming the naive baseline.

## Possible Further Improvements

Although a thorough effort was taken, there are ways to improve the predicting model for podium results in F1 races:

### Domain Expertise:

A greater understanding of the Formula 1 racing industry would be very helpful. Better feature engineering would benefit from deeper understanding of the importance of specific variables, such as track-specific features or driver situations.

### Addition of More Data:

Although the existing dataset covers a range of racing-related topics, adding more data sources, such as current weather information, tire wear rates, and even driver psychological conditions, could improve the model's predictive capability.

### Feature Engineering:

Improved performance can be attained by continuously growing and improving the set of features used in the model. It is possible to find hidden patterns by experimenting with combinations of existing features or by creating brand-new features from the given data.

## Conclusion:

In conclusion, it has been insghtful to follow the path from comprehending the value-based challenge to developing and accessing a prediction model for podium results in F1 races. The stated enhancements present chances to enhance the model's accuracy and broaden its applicability in the fast-paced and intense motorsports industry. The continual interaction between data analysis and racing knowledge will surely produce increasingly more precise and worthwhile insights as this endeavour continues.