Predicting Formula 1 Champions from Historical Data

Carlos Vega

April 9, 2024

1 Introduction

Formula 1 is regarded as the most prestigious racing series, the pinnacle of speed, power, and skill. It features the most advanced cars on the planet, driven by the most skilled drivers in the world. The top manufacturers compete against each other for a grand prize, status, and fame. For reference, the prize pool for the 2022 series was approximately 1.157 billion dollars, as the contract governing the series states that the team prize pot makes up approximately 50% of Formula 1's commercial rights profit. This creates an obvious incentive for teams to gain an advantage in every way possible, from scouting drivers to hiring the best engineers, etc.

Red Bull, for example, has finished on the top podium (either first, second, or third place) in every season consecutively since 2016, winning multiple Constructors' Championships and Drivers' Championships. But this was not always the case. Red Bull entered Formula 1 in 2005, established after the energy drink company Red Bull bought the Jaguar Racing Formula 1 team at the end of 2004. Red Bull Racing made its Formula 1 debut in the 2005 season with an initial driver lineup consisting of David Coulthard and Christian Klien. Coulthard, a seasoned driver with experience and race wins from his time at McLaren, was paired with Klien, a younger and less experienced driver coming from the Jaguar team.

Red Bull Racing's early years were about building a competitive team and establishing themselves in Formula 1. In their debut season, the team showed promise with several points finishes, and David Coulthard secured their first podium with third place in the 2006 Monaco Grand Prix. This win was significant as it marked one of the early successes for Red Bull in Formula 1, highlighting the team's potential and helping to establish its reputation in the sport.

In Formula 1, there are many ways to score points as a driver, both for oneself and for the team. Since 2010, a total of 102 points can been awarded to the first ten finishers, with 25 points for a first-place finish, 18 for second, 15 for third, then 12, 10, 8, 6, 4, 2, and 1 for the remaining finishers in the top ten. If a driver in the top 10 achieves the fastest lap, they score an additional point. This metric is what drivers compete for, who can collect the most amount of points by the end of the season. The performance of the drivers can give us insights into who is most likely to score most points. A natural question to ask is, is there a way for us to anticipate this placements, if so, could we run simulations to determine who would be the best driver that season?. Questions of this nature not only benefit fans, but also teams to see who are the drivers they should strive to acquire. In section 2 we will discuss the definition of phenomena and models and continue to built up on this idea of ”modeling”.

2 Phenomena and Models

A phenomena is anything that occurs in the natural world, weather patters, traffic, trains delays e.t.c In our case the phenomena is final points scored by drivers. In order to predict the future, one can develop a model for it, which for us is an approximation of reality. If out phenomena was to be train delays, we could create a model train system to see the underlying reasons that would result in delays. Immediately there is a problem, what causes delays?. Factors like train conductor being on time, accidents, policy intervention, fires in tracks and more. These factors still have fundamental issues, they are very ambiguous, accidents can occur for multiple reasons, people falling in the tracks, getting injured, needing medical assistance and many more reasons. Alternatively, we could take a more mathematical approach, this would allows us to remove any ambiguity, for example we can quantify the amount of police appearances in the train per week, this can give us a good measure of how often crimes/accidents can occur. Now bringing this back to out phenomena, we can measure the time per lap of drivers across all Grand Prix, this is a mathematical approach as there is no ambiguity. It general the *response measurement or output metric* is denoted mathematically as y, such that $y \in \mathcal{Y}$ the measurement of all possible output measurements. For us we are going to determine y to be final score of driver.

3 Driver Performance as a Function

For a better understanding it would be beneficial to think of out model as a magic box. The model is a magic box that takes in *inputs*, combines them in some way via ”magic” and outputs how many points the drivers will have by the end of the season. So our goal is to define this magic. One example of the inputs that we can use is the performance of the driver. This again leads us to the same problem of ambiguity, what does performance mean? and how can we measure it. For this problem we resort to mathematical models. *Mathematical models* have the advantage of they remove ambiguity just like our output metrics. We want to restrict the type of inputs to be only ones that can be represented with mathematics. This helps us remove ambiguity and makes it accessible to others who can help us evaluate the validity of our model. With all this being said, lets start developing the model. We first assume that our phenomena is deterministic.

$$y = t(z_1, z_2, \ldots, z_q)$$

This function is illustrating that the total points earned for a driver is determined by a "magic box" which takes in several *casual drivers* a.k.a the $z's$, combines them using magic denoted by t. Although we are able to determine these causes in a mathematical form, the complexity of the problem doesn't decrease. To better understand this complexity lets lost some reasonable casual drivers for our phenomena.

$$z_1 = \text{weather conditions in the track}$$

$$z_2 = \text{crashes in the race}$$

$$z_3 = \text{number of pit stops}$$

$$z_4 = \text{final position}$$

It is obvious how these casual drivers can be measures but they still fundamentally have problems. Although it is not ambiguity we struggle with, the problem is that they are unknowable to us. For instance, we are unable to use Z_1, to predict if the weather is going to be dry or wet its a challenge in itself and inaccessible for us. Therefore we resolve this issues accepting that function t is impossible to obtain. Therefore we to make another "magic box" f to approximate t. Even further we need to find approximations to our casual drivers to ones that can be measured prior to the season we want to predict. We let x_1 , x_2, …, x_p's be proxies/approximations for our z_1 , z_2, …, z_p's and define a new function f which takes in theses new x's as inputs. With all this in mind, let y now be represented as

$$y = f(x_1, x_2, \ldots, x_q) + \delta \textbf{ where } \delta \mathbf{= (}t \textbf{ - } f\mathbf{)}$$

These new stated inputs x_1 , x_2, …, x_p are often referred to as *features or predictors* and the term δ is known as "ignorance due to error". This "ignorance" is related to our impossible to get casual drivers and how far off our approximations are. Some possible features we can use in our new f model are could be the following

$$
\begin{aligned}
x_1 &= \text{Average lap time in Grand Prix}\\
x_2 &= \text{Average Speed of fastest lap in previous years}\\
x_3 &= \text{Starting Position in Starting Grid}\\
x_4 &= \text{Average Points at the end of the race based on grid position}
\end{aligned}
$$

These new introduced x's are features which are avalible to us prior to each race and in adition of that, they are numerics. With these factors in mind it is very intuitive how they can give us an insight in how a particular driver. If they have a starting position in the front, and they average a faster speed than its neighbors in the grid, they are likely to end in a higher rank. These proxies arent perfect and there are numerous other proxies that could help us like the "inteligence" of the driver, how good he is at taking corners, making decision, team plans of how to engage in the beginning of the race. But again they are impossible to quantify. There are also alot of "garbage" proxies we can use, Previous Season's Team Standings. at first this seams like a good candidate so why would we consider this garbage. A team can change fundamentally from one year to another, as explained earlier Red Bull previously was Jaguar Racing, as the years go by, engineers, car design, drivers, management all change. So adding unrelated features can cause our predictive measurements (RMSE) to be off by huge factors. So how do we pick the best proxies? We will be explore this in further sections.

4 Learning From Data

Even with all this already set there is an underlying issue we have not tackled, how are these features related i.e what "spell" are we going to use in out magic box to combine these features? The approach we use in this problem is something called Supervised Learning from Historical Data.

What is Supervised Learning from Historical Data? *Supervised Learning* is a method in which models are able to "learn" from data in order to find a function that can best relate the input measurement; our x's, to our desired output y. The desires function is one found by using this historical data and through some criteria and algorithm of our choice, infer that best represented the association of these features to best predict our outputs. In a mathematical notation, our new model g is the function created from this algorithm $\mathcal{A}$, over subset of candidate functions $\mathcal{H}$ and our historical data D i.e let $g = \mathcal{A}(\mathcal{D}, \mathcal{H})$.

4.1 $\mathcal{H}$ - Set of Candidate Functions

So far we have attempted to stay as true to the original phenomena represented as

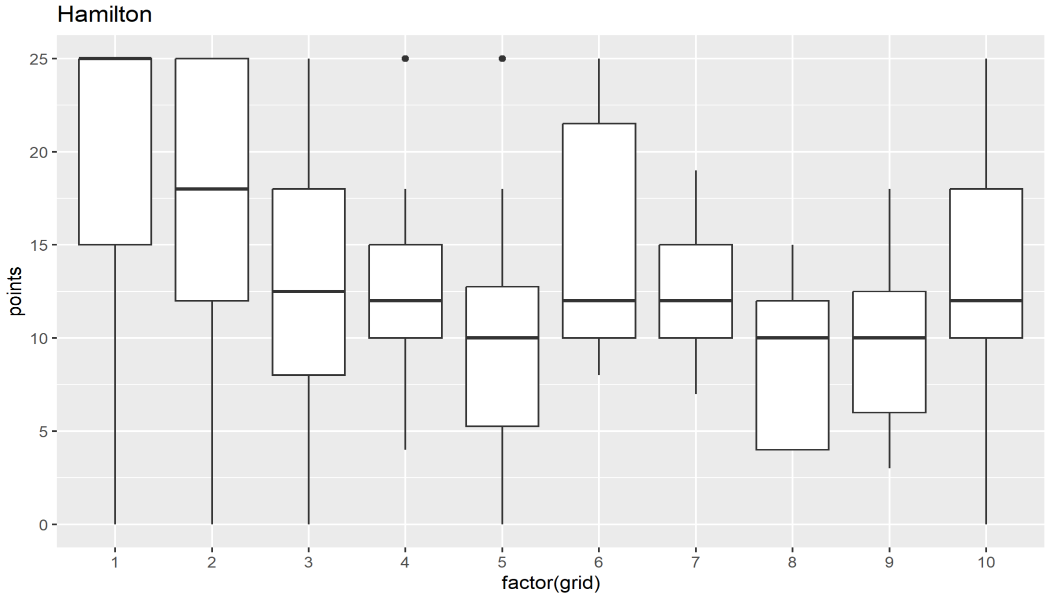
$$y = t(z_1, z_2, \ldots, z_q)$$

We made substitutions to this function to be able to approximate these unobtainable casual drivers with the following function

$$y = f(x_1, x_2, \ldots, x_q) + \delta$$

Our biggest enemy is the unknown complexity of the function that can describe our phenomena, this is an enemy we will never be able to beat since the set of all functions ($\mathcal{H}$) is incomprehensible complex. Therefore it is necessary to reduce our set of functions to make reasonable within our approximation necessity. What $\mathcal{H}$ would look like is based on our interpretation of the relation between all our features. I believe a linear model would be best for approximating our relationship. Therefore for us this subset $\mathcal{H}$ is denoted to be all linear functions.

Lets examine some of the connections between our features, take x_3 and x_4 for example, x_3 is the starting position in starting grid of a driver. In the graphic below we see the average points scored at the end of the race for each starting position.



We can see a trend forming, when Hamilton started in first place the race, on average he finished in first place and was awarded 25 points. Even on the other side of the spectrum, when Hamilton started on 10th place, he would average 10 points which is the points awarded to 5th place. For a really good driver as himself there seems to be a relationship on how well he will perform at the end of the race based on his staring position. It is easy to visualize a line that would approximately go through the averages, We can represent this relationship mathematically as follows.

$$h(x) = b_0 + b_1 x_1$$

If we restrict our model to using 1 of these x which is the average of points, our new candidate set would be the following

$$\mathcal{H} = \{w_0 + w_1 x_1 : w_0, w_1 \in \mathcal{R}\}$$

this is, all possible single polynomial lines that show the relationships when you pass the average points per starting position. Furthermore we assume the existence of a "line of best fit" for both of these proxies, let $h^* \in \mathcal{H}$.

$$h^* = \mathcal{B}_{\prime} + \mathcal{B}_{\infty} x_1$$

where $\mathcal{B}_{\prime}$ and $\mathcal{B}_{\infty}$ are our best approximations to the real coefficients. Our algorithm will essentially search for the best approximation h^*, and this approximation will result in our previously mentioned g.

Now that we have an idea of what our function looks we should correctly modify the set of all possible candidates to something more precise. We want to generalize our $\mathcal{H}$ to be more than just a line but for the addition of more relationships this $\mathcal{H}$ can be a hyper-plane (a plane in higher dimensions)

$$\mathcal{H} = \{b.x : w_0, \ldots, w_4 \in R\} = \{w_0 + w_1 x_1 + \ldots + w_4 x_4 : w_0, \ldots, w_4 \in R\}$$

4.2 Training Data $\mathcal{D}$

To train our model we should have a systematic way of inputting the same type of information to our model, if we try to compute our own line to give us our closest approximation to h^* we need to pass our measurements $x_1, \ldots, x_4$ as well as the final points awarded to our the driver, y. We represent this in a vector format as follows:

$$[x_1 = 90091, x_2 = 206.0523, x_3 = 1, x_4 = 25|y = 25]$$

In words, Hamilton had completed on average each lap in 90091 milliseconds (chose this scale since my dataset had other metrics also in milliseconds), on average his speed on his fastest lap was 206.0523 miles per hour, started in pole position (first position) and ended with 25 points on average. Now imagine repeating this process for every race our driver has participated in his professional career. Hamilton has been driving for Formula 1 since 2006, we can represent all this data in a matrix where n = 23 * (2023-2006) =¿ 23 * 391; since there are 23 grand prix a year and 5 columns.

$$\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{391,1} & x_{391,2} & x_{391,3} & x_{391,4} & x_{391,5} \end{bmatrix} \quad \begin{bmatrix} y_1 \\ \vdots \\ y_{391} \end{bmatrix}$$

Why add a column of 1's at the beginning of the matrix? In the context of linear models, adding a column of 1s as the first column of the design matrix is essential for including the intercept term in the model. This column allows the linear model to have a non-zero y-intercept, which is critical for accurately fitting the data, especially when the relationship between the variables doesn't pass through the origin. We also separate both inputs and outputs into 2 matrices, $\mathcal{X}$ and y. Matrix $\mathcal{X}$ is also as out *training data*, we denoted $D =< X, y >$. With this being said and also having our $\mathcal{H}$ and D we are only missing the "spell" to combine all of these data points. This algorithm, lets call it A, will be discovered through supervised learning.

4.3 Algorithm possibilities for A

An algorithm is a step-by-step procedure or a set of rules to be followed in calculations or other problem-solving operations. For our case out the problem is finding our model g, the approximation of h^*, which is the approximation of f, a simplification of the unknown t. What our algorithm will attempt to do is to return the best guess $\hat{b} = \{\hat{b_0}, \ldots, \hat{b_p} \in R\}$ of the coefficients for our possible candidate functions previosly mentioned. We have multiple algorithms we can choose, for this case we will use something called OLS (Ordinary Least Squares). It is one of the most common methods used in the context of linear models to find the best-fitting line (or hyperplane in higher dimensions) that minimizes the sum of the squared differences (residuals) between observed values (dependent variable) and the values predicted by the linear model.

4.3.1 Ordinary Least Squares

Lets try to understand what ols is trying to do, we will limit our example to a 3D space for the sake of visualization. Imagine that we have 2 features, x_1 and x_2 that best describe our y. If we were to graph all the observations, in a a classic manner there is a way to fit a line of best fit. Our specified $H = \{\text{All linear functions}\}$ would represent all possible lines that could fit this model, if we add a third feature this would be instead of a line a 3-dimensional plane instead. Our algorithm runs on our data set $A(D, \mathcal{H})$, and would attempted to fit the best hyper-plane to this data set, but what do we mean by best?. We can use a objective function, a mathematical relation that our algorithm attempts to minimize (or maximize depending on your goal), the result then is the "best" choice given our constrains. For us the objective function we try to minimize is something called SSE or sum of squared errors.

SSE as the name implies is the sum of the squared errors from the predicted function $\hat{y_i}$ and the actual data points y_i. Our algorithm is taking the difference between them $y_i - \hat{y_i}$ which is the error of the current estimate. But why square? We square to make sure the error is non-negative, also by squaring the bigger errors would hold more weight in our prediction make this beneficial. After computing all the summed squared errors our algorithm will return the min vector prediction of all the coefficients that best fit this plane to the dataset. In mathematical notation:

$$\hat{b} = arming_b ||y - \hat{y}||^2$$

This makes sense, if what the algorithm is trying to do is find the plane that best fits the data, you want that fit to be the plane that has the least error, or the smallest distance to each point for a better prediction.

Something really interesting is that the plane that results from this minimization would be one where each prediction of $\hat{y_i}$ is an orthogonal projection of y_i onto a plane defined by a linear combination of our features. This in a mathematical context means that the line representing the error $y_i - \hat{y_i}$ forms a right angle with the plane and thus forms the shortest path between each y_i and the corresponding $\hat{y_i}$.

4.4 Other Sources of Error

So far we have only not been specific forms of error, we have interpreted error as the difference between each data point and our model. This error is also exclusive and not generalized for models with different algorithms and different purposes. Our goal is to generalize possible errors in order to address them one by one. Earlier we explained that the true casual drivers need to be approximated using proxies, these drivers are fundamentally unobtainable to us due to the complex relationship between them, This is where we encounter our first error, *ignorance* error denoted as $\delta = (t(z) - f(x))$. In essence this error stems from the magnitude of difference between our estimation and the real causal drivers. We defined $\mathcal{H}$ to be the constricted set of possible candidate functions that would predict our phenomena, these functions can; and most often; are extremely complex in nature. If our phenomena is complex, i.e stock prices, a simple ”price over time” linear model will not be able to explain it due to its simplicity. This type of error is called misspecification, denoted as $(f(x) - h^*(x))$. Where h^* is the best candidate function in our set of candidates and f is our best approximation of h^*. At last, for the case of this phenomena, driver performance, if we try to train our model in just 1 years worth of data there will be alot of information left behind, maybe that year is an outlier, or maybe that year was the peak. We remove the relation as a whole due to a lack of observations. This type of error is called *estimation* error, denoted as $g(x) - h^*(x)$.

We can sum up all of these errors and apply them in a function.

$$y = g(x) + (g(x) - h^*(x)) + (f(x) - h^*(x)) + (t(x) - f(x)$$

5 Error Metrics

Despite the improvements we have made producing a model that can approximate our phenomena, there are still many things we need to consider. Imagine that the model is giving us garbage despite our best attempts to add good, relatable proxies. In the other hand, what about the opposite scenario where provide garbage proxies but yet it appears that our model is working just fine. We need a way to gauging our predictive power.

A prediction metric is a way for us to accurately measure the predictive power of our model. In terms of this case since we have used OLS as our algorithm there are preset metrics for it. Two of the most used metrics are R^2 and RMSE. Let R^2 and RMSE be defined by the following functions.

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$RMSE = \sqrt{\frac{1}{n-p-1}SSE} = \sqrt{\frac{1}{n-1}\sum_{i=1} n(y_i - \hat{y_i})^2}$$

RMSE, or Root Mean Squared Error, measures the typical or average distance between the predicted values and the actual observed values. We square them to return the error back to the original units of the errors. SSE is the summed of squared errors. The errors are squared to make them all positive and also weights the greater errors more due to the nature of exponential.

6 Overfitting Underfitting

In predictive modeling, methods like RMSE and R2 may sometimes cause overfitting while trying to minimize error. Yet, assessing models through these metrics remains crucial. Overfitting happens when a model becomes too intricate, catching data's noise instead of real trends. This leads to great results on training sets but poor ones on new data. Picture a student who memorizes answers without understanding the concepts; they might do well initially but falter with new problems. In Formula 1 predictions, overfitting could make a model overly specific to past race data, harming its ability to predict future performances accurately.

Adding too many features or variables can trigger overfitting. Although it seems helpful at first—potentially lowering RMSE—it might just add noise and promote overfitting. Take predicting football passing touchdowns as an instance: Including irrelevant factors like game day weather or the player's preferred color could falsely boost the model's accuracy without really improving its prediction power. Much as cramming useless info into a study guide muddles rather than clarifies, so does packing models with unrelated details confuse rather than enlighten on the topic at hand.

In any analysis, one should consider how feature quantity stacks up against data instance count. With more traits nearing the volume of data bits, overfitting danger climbs. This scenario is akin to cramming too many jigsaw pieces into a small frame; what results fails to capture the full image accurately. Within Formula 1 forecasting realms, this could translate into an excessive reliance on old figures without weighing the sport's evolving aspects and variables impacting driver performance.

Moreover, overfitting not just skews model efficacy assessment but also curtails its predictiveness with fresh inputs. Hence, even with high success rates in training phases, models might falter in foreseeing outcomes for future episodes. For Formula 1 scenarios specifically: A model prone to overfitting may show prowess in decoding past race results through historical lenses yet stumble when tasked with projecting effects from shifting track states or driver game plans on looming competitions.

Wrapping up; grasping overfitting's drawbacks proves pivotal within predictive modeling territories - more so amidst complex and fluctuating fields like Formula 1 racing contests. Awareness and caution towards overfitting hazards pave paths toward devising forecasts that are both precise and dependable.

7 Model Validation

Model validation is a crucial step in constructing mathematical models to accurately predict driver performance in Formula 1. By partitioning historical data into training and testing sets, denoted as

Dtrain and Dtest, we can ensure the model's accuracy. This partitioning is done randomly to avoid time-related patterns, with a predetermined proportion K. The model is then trained on Dtrain to create gtrain, which is used to predict outcomes on the 'new' test data in Dtest. This process allows for the computation of out-of-sample error metrics, such as RMSE, which provide an upper bound on the model's generalization error.

When constructing the final model, gfinal, the entire dataset D is used to ensure no usable data is left behind. However, this process relies on the assumption of stationarity, meaning that the underlying drivers and phenomena remain consistent over time. Without stationarity, the modeling process becomes unreliable. In the context of predicting driver performance in Formula 1, stationarity is crucial to ensure the model's effectiveness in making future predictions.

7.1 Importance of Honest Error Metrics

Obtaining honest error metrics through model validation is essential for evaluating the accuracy of predictive models. By partitioning the data and computing out-of-sample error metrics, researchers can assess the model's performance on unseen data. This approach helps in identifying potential biases or overfitting issues that may affect the model's predictive capabilities. Honest error metrics provide a realistic assessment of the model's generalization error, allowing researchers to make informed decisions about the model's reliability.

For example, in the field of finance, accurate prediction of stock prices requires robust models that can generalize well to new data. By validating these models using out-of-sample error metrics, analysts can determine the model's effectiveness in real-world scenarios. This process ensures that the model's predictions are reliable and can be used to make informed investment decisions.

7.2 Challenges and Considerations in Model Validation

While model validation is a powerful tool for assessing predictive models, it comes with its own set of challenges and considerations. One of the key challenges is ensuring the stationarity of the underlying phenomena, as any changes over time can impact the model's accuracy. Additionally, selecting an appropriate proportion for partitioning the data into training and testing sets is crucial to obtain reliable error metrics.

Moreover, the choice of performance metrics used to evaluate the model's accuracy plays a significant role in model validation. Metrics such as RMSE provide insights into the model's predictive capabilities and help researchers fine-tune the model for better performance. By considering these challenges and factors, researchers can enhance the reliability and effectiveness of predictive models in various domains.

8 Model Selection

Selecting the optimal model in predictive analytics involves navigating the balance between complexity and accuracy, to avoid overfitting while ensuring sufficient explanatory power. Given the numerous challenges associated with model validation, the selection process becomes crucial for predictive success, especially in dynamic environments like Formula 1 racing. Model selection should be driven by criteria that encapsulate both predictive accuracy and model simplicity. The model should have low out-of-sample error, as indicated by metrics like RMSE or MAE, ensuring it generalizes well to unseen data.

8.1 Simplicity

Prefer simpler models that achieve similar predictive accuracy to more complex ones, adhering to Occam's Razor principle. This reduces the risk of overfitting and improves model interpretability. Employ k-fold cross-validation to assess model performance across different subsets of the data, enhancing the reliability of the performance estimates. Feature selection plays a pivotal role in model selection, identifying the most relevant features that contribute to predictive accuracy without introducing unnecessary complexity.

8.2 Cross-Validation Performance

Techniques like stepwise regression, lasso, and ridge regression can help in striking a balance between including informative features and avoiding overfitting. Models are evaluated based on their performance on a separate validation dataset, with the model having the best cross-validation score typically being selected. The decision process involves comparing models based on the chosen criteria and selecting the one that offers the best trade-off between complexity and predictive power.

8.3 Feature Selection and Interpretability

Understanding the factors influencing outcomes is crucial in contexts like Formula 1, where model interpretability is as important as prediction accuracy. Once a model is selected, it should be validated using a fresh dataset not used in the training or selection process, ensuring the model's performance is genuinely representative of its ability to generalize. This final validation confirms the model's predictive capabilities and its readiness for practical application, selecting the optimal model in predictive analytics requires a delicate balance between complexity and accuracy. By prioritizing simplicity, employing cross-validation techniques, and focusing on feature selection and interpretability, a robust predictive model can be developed. The process of model selection is essential for ensuring predictive success, particularly in dynamic environments like Formula 1 racing. By following these guidelines, organizations can make informed decisions based on reliable predictive models.

9 Conclusion

Formula 1, a sport defined by speed, tactics, and precision where trying to predict success necessitates not merely numbers but skillfully amalgamating information with how circumstances can evolve throughout races. This paper tried to make sense of all the different factors that affect results in Formula 1 by building a predictive model based on past records but that could still adjust to what happens on the track, even if plans change.

While exploring ideas and models, we dug deeper than just how drivers do, diving into math stuff to find some kind of order in the mess. Like it talks about in Section 2, we found out racing isn't just about one thing, as weather, crashes, and plans all mix together to shape each race and the whole season.

Like it says in more detail in Sections 3 and 4, it wasn't just connecting dots but figuring out Formula 1's "heartbeat". Through a meticulous, step-wise methodology in constructing our models, with contemplation given to the various factors and their interplay, as well as confrontation of the reality that certainty remains elusive in this unpredictable arena, we advanced in our work. Looking at all this showed how complicated it is to truly capturing a driver using numbers.

Upon arriving at the section concerning machine learning techniques, the author illuminated some of the difficulties surrounding selection of the most appropriate algorithm as well as potential issues that may arise from over-engineering to training data. The discussion around Ordinary Least Squares and other algorithm possibilities highlighted how hard it is to balance a model's complexity with how accurately it can predict. Our story, told through Section 5, emphasized how important error metrics are for seeing how reliable our prediction attempts were.

They checked over how they figured stuff out real close, making sure their way of thinking stayed with what's real and didn't get tricked by seeming like it worked too well just for what they learned from. When they started counting mistakes too, it let them see actually how good their thinking was instead of just on the things used to train it.

In the end, they agreed on choosing what to go with by marrying easy with complex, making their way of thinking not just some math thing but also a tool that can deal with how you can't predict Formula 1.

Overall, predicting who would win Formula 1 titles from history teaches the togetherness of figuring stuff out and how sports can surprise. As we look ahead, with ways of thinking made better by what happened before and proven by strict ways of doing it, we're on the edge of turning just numbers into insights, guesses, and ultimately a deeper understanding of both the art and science of Formula 1 racing.

10 References

Dataset. https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

Formula 1 results. https://www.formula1.com/en/results.html/2024/races.html