Predict if a driver will score points in an F1 race

Mariana García Badillo A01272957
Tecnológico de Monterrey, Campus Querétaro

***Abstract*- This document presents the implementation of machine learning algorithms such as Random Forests and Logistic Regression and their results with predicting if an F1 driver will score points in a race.**

I. INTRODUCTION

Formula 1 since its beginning in 1950, has been evolving and is for sure, one the sports where technology has seen the greatest developments, from the more of 200 sensors generating 3GB of data each race, to the hybrid motors era. There are a lot of things and details to have in consideration when talking about Formula 1 cars, engine performance, tire grip, telemetry, fuel consumption, and others. Engineering teams need all this information to understand the kind of strategy they will use during a race weekend. To help them with this, teams are starting to use different IA algorithms to help them keep track and develop better strategies. As that information is private per team, the only data available to public are race results with information like, initial position, fastest lap, constructor, driver, pit stops for each race, etc, which can be used to make different predictions such as how many pit stops per races can be done, how many non finishers, among others. In this paper, the focus will be in predicting if a driver with a constructor might finish in the top 10, giving the driver an amount of points that help him and the team on the drivers and constructors championship; this done with two different Machine Learning algorithms.

II. STATE OF THE ART

Random Forest is a kind of supervised learning algorithm and is known to give great results for both classification and regression tasks. The way it works is by building and ensemble of decision trees that will be trained by the bagging method to make a combination of learning models to increment the result and obtain a more accurate prediction. Another thing to consider with Random Forest is the randomness it gives to the model while growing the trees, this allows the model to search for the best feature of them all to make a better prediction. [1] Figure 1 shows a simple architecture of a Random Forest Classifier.

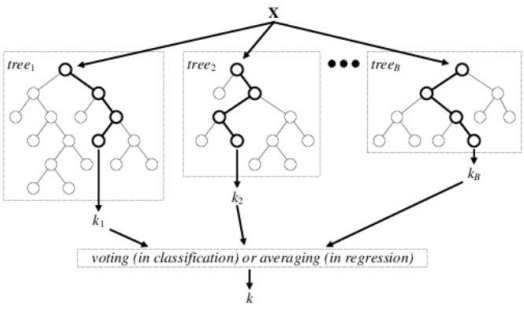

Figure 1. Random Forest architecture

Logistic Regression is another common method used for classification problems in Machine Learning and just like Random Forests, is a supervised learning algorithm. This model is most used when the value target variable is categorical in its nature and when the target data is a binary output, meaning if it belongs or not to a class. However, this does not exclude the algorithm to work well with multiclass classification problems.[2] The basis of logistic regression is, as its name, the logistic function, or the sigmoid function, which takes in any real value number and maps it to a value between 1 and 0.[3] Figure 2 shows the formula that is implemented when using Logistic Regression. In Figure 3, an example of this algorithm can be observed. The particular S figure marks the boundary between the true samples and the false one.

$$Sigmoid\ Function:\ y = \frac{1}{1 + e^{-x}}$$

Figure 2. Logistic Regression function

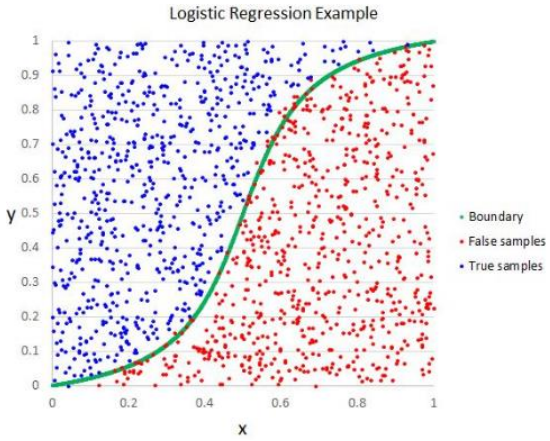

Figure 3. Logistic Regression example

III. DATASET

The data set was obtained from Kaggle [4], and it contains the results for all races since its beginning in 1950 to the first race of 2021 in Bahrain, meaning there are more than 20,000 entries. This is a series of datasets that are related through Ids; the ones that are used for this project are the datasets results, races, drivers, constructors, and circuits. Some preprocessing had to be done to have the data in a final dataset. Datasets results and races were merged into one, dropping the attributes that were not needed, leaving a dataset with only the following attributes: constructorId, driverId, grid (initial position), position (final position), year and circuitId. From this attributes, null values were also dropped as there was no other option possible, if assigned a mean value, the results would not be correct as from 24,985 values, around 10,000 were null values. These null values correspond to either not taken place on the qualification session to decide the starting position or to not finishing the race. There are several reasons to not finish a race, engine problem, brake failure, crash, however this data was not available. From previous knowledge and research, there have been changes over the number of drivers per race, from 10 to 34. In the last 10 years, a grid of 20 drivers has been decided, meaning that the ones starting from lower positions, would be missing, so, to manage this data, the ones with a starting position higher than 20, were rounded out as 20. And last, a new attribute was added to the final dataset, this validates if a driver has finished in a position from 1 to 10, this awards an amount of points to the driver and constructor, allowing the project to predict with a 1 if the driver has points and a 0 to not finishing top 10. This new attribute will be the target of the model. After this, the final position attribute was dropped as it was not needed as an input.

	constructorId	driverId	grid	year	circuitId	score
14205	3	849	18	2020	24	0
14206	210	825	20	2020	24	0
14207	210	850	17	2020	24	0
14208	131	1	2	2021	3	1
14209	9	830	1	2021	3	1
14210	131	822	3	2021	3	1
14211	1	846	7	2021	3	1
14212	6	844	4	2021	3	1
14213	1	817	6	2021	3	1
14214	6	832	8	2021	3	1
14215	213	852	13	2021	3	1
14216	117	840	10	2021	3	1
14217	51	8	14	2021	3	0
14218	51	841	12	2021	3	0
14219	214	839	16	2021	3	0
14220	3	847	15	2021	3	0
14221	117	20	20	2021	3	0
14222	210	854	18	2021	3	0
14223	213	842	5	2021	3	0
14224	3	849	17	2021	3	0

Figure 4. Last 20 values of the final dataset

IV. MODEL

As is a classification problem, both algorithms are good options to solve the problem. As this is taking into consideration different variables, the best option is to choose the Random Forest. After running a GridSearchCV, the best results for the hyperparameters to use in the the model were: max_depth=25, min_samples_leaf=1, min_samples_split=15 and n_estimators=1200. However, the Logistic Regression algorithm works best with a max_iter=9000, if this number gets higher, the accuracy of the model descends, and when reducing this number, the model is not capable to run, as the value is not enough for the data.

V. RESULTS ANALYSIS

Both implementations were made using frameworks and as mentions above, some hyperparameters were tuned to obtain the best

outcome as possible. As both algorithms have been implemented, the best way to compared them was with the accuracies scores.

The Logistic Regression throws an average accuracy of 76.733%, while the Random Forest throws an average accuracy of 81.300%.

It can be seen that both models have a good percentage of accuracy, however, Random Forest is the algorithm that after some testing, gave more realistic results. For example, a Williams car in the last years has not been able to end on points positions when starting from 14, a feasible position to end on points, the Logistic Algorithm predicted it would end top 10. However, the Random Forest classifier predicting it would not end on points.

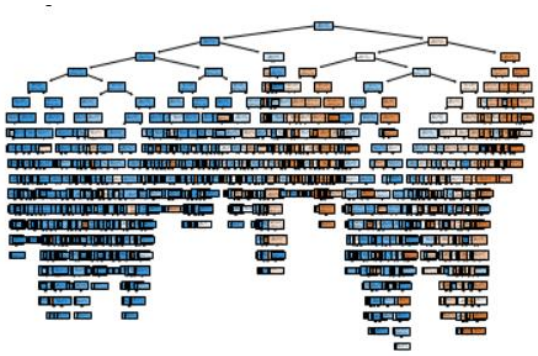

Figure 5. Single tree from the Random Forest

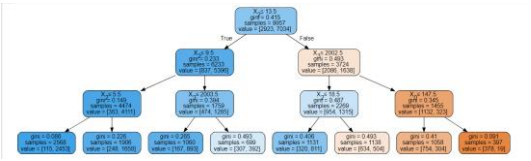

Figure 6. Classification tree with the train data

VI. CONCLUSIONS

One thing to have in mind, F1 races depend on way much more that the attributes taken into consideration. Weather, pit stops, safety cars, laps per circuit, tire strategy are some of these things that can change the race in one second. Another important consideration, this data does not contain the reason for a driver to not finish the race. If this data was easily available, it might be able to help make a better prediction, as it would have a record of the teams technical failures and the driver crashes, giving more statistics and values to consider. To make predictions in sports is tough, as there are variables that can change the whole situation in a matter of seconds, however, using IA to predict certain scenarios is useful as it can help teams plan a strategy with past data.

VII. REFERENCES

[1] Donges, N. (2019). A Complete Guide To The Random Forest Algorithm. Retrieved from https://builtin.com/data-science/random-forest-algorithm

[2] Kambria. (2019). Logistic Regression For Machine Learning and Classification. Retrieved from https://kambria.io/blog/logistic-regression-for-machine-learning/

[3] Yildirim, S. (2020). How is Logistic Regression Used as A Classification Algorithm? Retrieved from https://towardsdatascience.com/how-is-logistic-regression-used-as-a-classification-algorithm-51eaf0d01a78

[4] Vopani. (2021). Formula 1 World Championship (1950-2021). Retrieved from https://www.kaggle.com/rohanrao/formula-1-world-championship-1950-2020?select=results.csv