Machine Learning Formula One

STAT 442 Final Project

Jack Forrest and Jimmy DeLano

May 16, 2022

Professor Ning

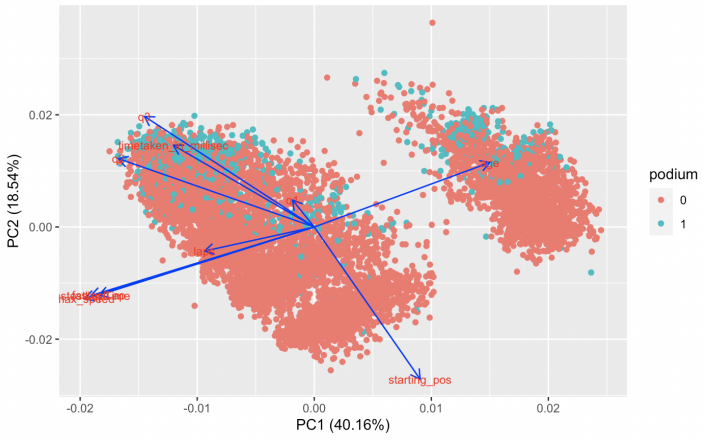
Introduction

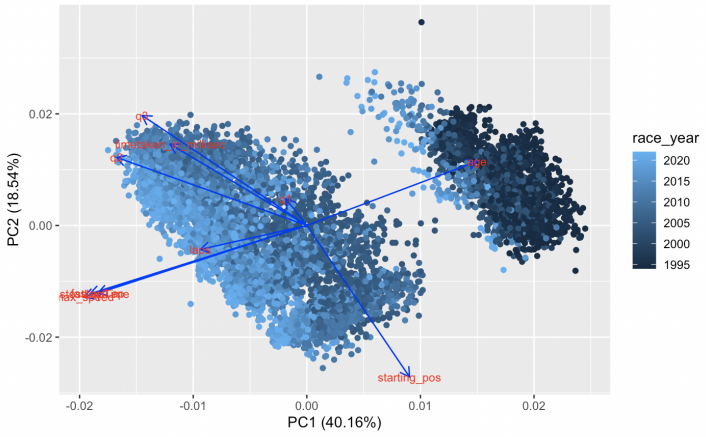
Formula One racing is an immensely popular global phenomenon. The recent Netflix show “Drive to Survive” brought the sport into American popular culture and has resulted in an uptick in the US Formula One fandom. We hoped to use historical data on Formula One races to develop models that could predict whether a race that fit certain criteria would result in a podium finish or not. This could potentially be applied to future race data to determine what variables are the most important in determining a successful race.

Data

Our data comes from the Formula 1 World Championship page on Kaggle (https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020) which contains historical records on the sport dating back to 1950. The original structure is organized into several tables. For this project, we used the ‘results’, ‘status’, ‘drivers’, ‘races’, ‘constructors’, ‘driver standings’, and ‘qualifying’ tables merged together. This repository contains a massive amount of data, so we narrowed down our focus to predicting whether or not a driver would finish on the podium for a given race. To do so, we created a new binary column where podium == ‘Yes’ corresponds to a top 3 finish and ‘No’ corresponds to a fourth or worse finish. The resulting dataset had 8954 rows and 18 columns. The features relate to the race itself (track, number of laps, year), the driver’s performance leading up to the race (# of constructor points before the race, qualifying times, starting position), driver demographics (age, nationality, company), and mid-race metrics (fastest lap time, max speed).

While exploring our initial dataset with PCA, we found that the data formed two distinct clusters. These clusters did not relate to the podium finish for a given race, as shown in the first PCA autoplot below. Rather, the two groups seemed to be explained by differences in the year of the race (second plot). One cluster corresponds to most races prior to 2005 and the other cluster to races after 2005. The sport has undertaken several major rule changes over the years which could explain this difference, but, regardless, we needed to account for this in our modeling process.





As a result, we decided to proceed with modeling two separate datasets: F1 races before 2005 and after 2005. In both cases, we split the data into training and test sets using a 70/30 split. The before data contains races between 1994-2004 and the after data contains races between 2005-2022. From here on, we refer to these as the “before” and “after” datasets. The before data did not have any information about 2nd or 3rd round qualifying so we dropped these columns which were all NAs. The last edit we made to the data was dropping the race_track, company, nationality, and driver_name columns. We ran into several downstream errors in several models where the test data had different levels for the factors as the training data. We were left with 11 numeric predictors for the before data and 13 numeric predictors for the after data.

Modeling / Results

Since we are hoping to predict a podium finish, we decided to use all the classification models that we have seen in class this year. We began with a basic logistic regression with all predictor variables to establish a benchmark model. From there we decided to use KNN, LDA, QDA, SVC, Trees, and Random Forests to predict a potential podium finish. As a result, we ended up with 9 models for each dataset. As a general baseline for all the models, 85% (17/20) of drivers in a given F1 race do not finish on the podium. Thus, a model that always predicts no podium would be right 85% of the time.

Let's start by looking at logistic regression for each of the two datasets to establish a benchmark model. Running a basic logistic regression with all the numeric predictors for the before dataset gave us an out-of-sample prediction accuracy of 91.9%. For the after data we got an out-of-sample prediction accuracy of 90.7% when using logistic regression. This is a relatively high prediction accuracy for the most basic model.

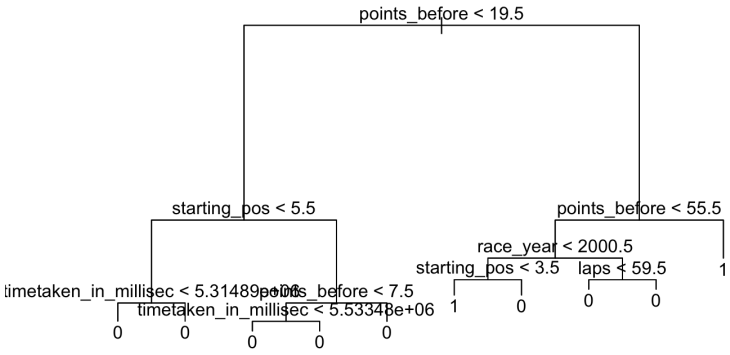
We then chose to run KNN on both datasets using 5-fold cross-validation to select the value of k and the kernel and then refit a model based on these parameters. For the before dataset, we found that KNN with K=5 and a gaussian kernel achieved an 88.9% out-of-sample

prediction accuracy. Running KNN with K=5 and a gaussian kernel gave us a 91.9% prediction accuracy for the after data. KNN performed relatively well across both datasets.

Next we ran LDA & QDA with uniform priors. LDA performed better for both datasets: we had a 85.1% prediction accuracy on the before dataset and a 88.1% prediction accuracy for the after dataset. The out-of-sample prediction accuracy for QDA was 81% for the before data and 79.9% for the after dataset. QDA was the worst model we looked at. As we saw through examples in class, these methods have strict assumptions and don't always work that well in practice. In fact, QDA is worse than just always predicting no podium.

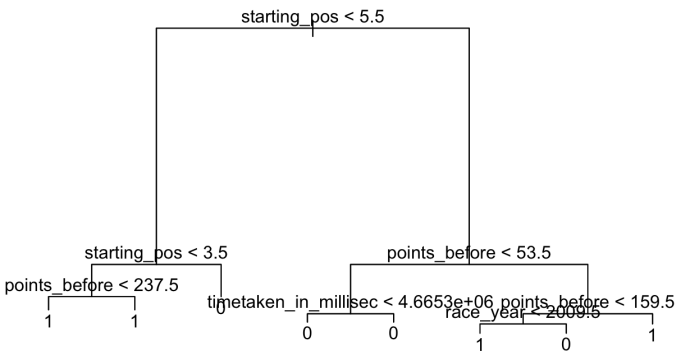
We then ran SVC on both datasets using 5-fold cross-validation to select the value of C, the cost or total slack allowed. For both models we used a linear (vanilla) kernel function. With c=0.1, the out-of-sample prediction accuracy was 88.7% for the before dataset. For the after dataset, the prediction accuracy was 91% using c=1.14. SVC performed relatively well on both datasets compared to the rest of the models we looked at.

Next, we decided to use Trees to hopefully improve the prediction accuracy and determine the most useful variables. We began with the most basic Tree model and then used both pruning and bagging to narrow the model down. When we used simple Trees on the before dataset we were left with the chart below.



We see that the model begins with points before and then splits the tree using other notable predictors including starting position and time taken. The most basic trees gives us a prediction accuracy of 87.4%.

We went through the same Tree process for the after dataset and were able to generate the chart below.

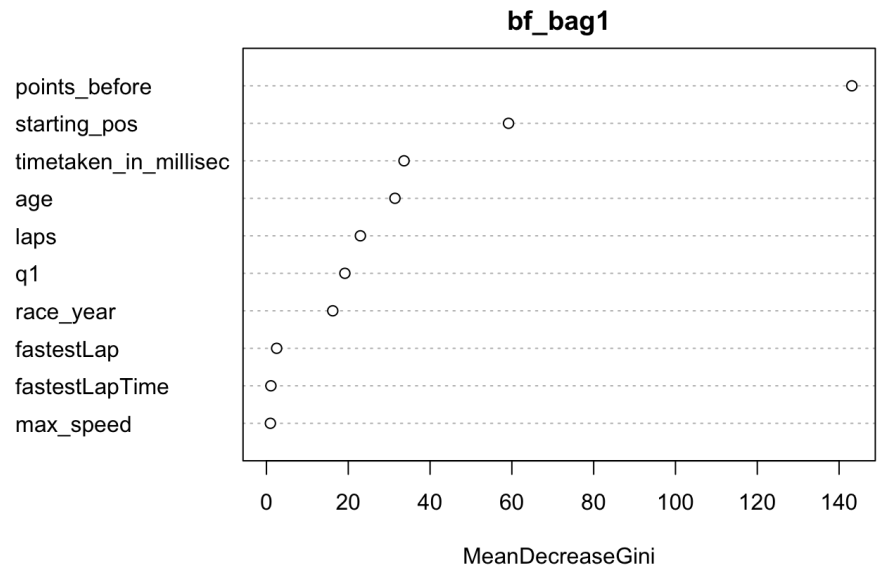


Here we see that the model follows a similar pattern to the model using the before data. We begin with the starting position, move into points before, and finish with time taken. This model gave us a prediction accuracy of 90.4%.

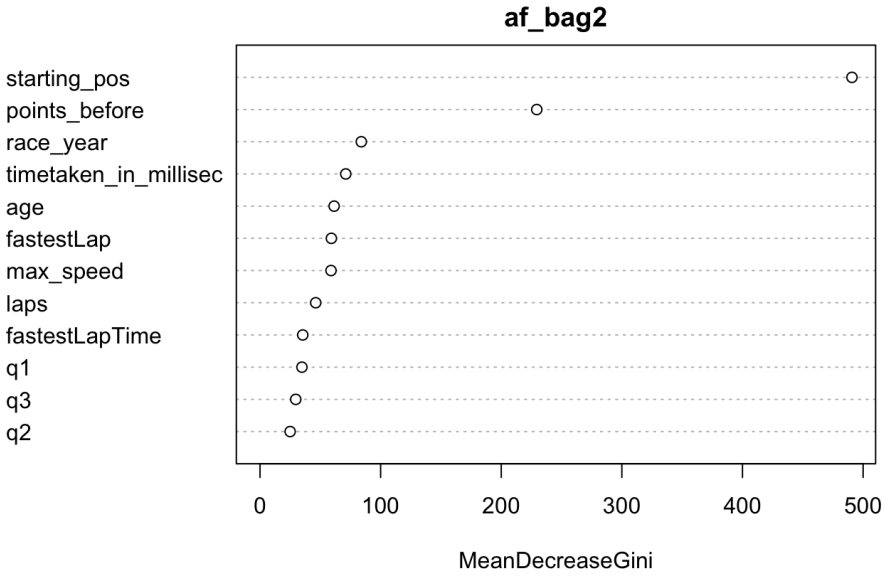
We continued by pruning the trees to hopefully improve the prediction accuracy. We pruned the trees by misclassification. Unfortunately, the prediction accuracy for the before dataset actually got worse when we pruned the trees, dropping to around 87.2%. When pruning the trees for the after dataset we received the exact same results as the non-pruned trees (90.4% classification accuracy).

Since decision trees performed relatively well we decided to use bagging to hopefully improve our prediction accuracy. These models performed quite well, giving us a prediction accuracy of 88.9% for the before data and 92.4% for the after data. The 92.4% for the after data was one of the highest prediction accuracies we would see. We also developed important variable plots for both bagging models. The before and after plots are shown below.

Before:



After:

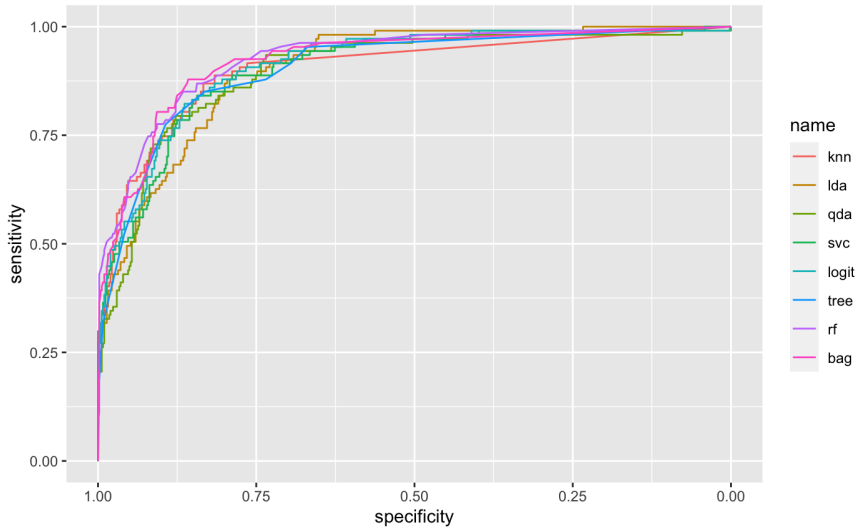


We see that in both datasets starting position and points before are the most important predictors.

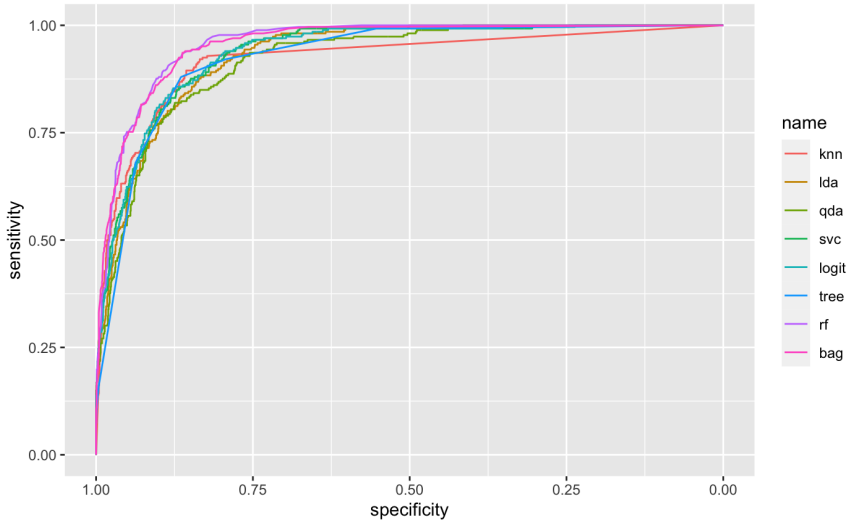
Finally, we used Random Forests, which performed very well for this data. The Random Forest model gave us prediction accuracies of 89.7% (before) and 92.8% (after). This is the second highest prediction accuracy for the before data and the highest for the after data. All models and their corresponding prediction accuracies as well as the ROC curves for both datasets are contained in the figures below.

	Before <dbl>	After <dbl>
KNN	0.8889	0.9195
LDA	0.8513	0.8810
QDA	0.8137	0.7990
SVC	0.8873	0.9104
Logistic	0.9199	0.9075
Trees	0.8742	0.9041
Pruned Trees	0.8725	0.9041
Bagging	0.8889	0.9239
Random Forest	0.8971	0.9282

Before:



After:



Conclusion:

Throughout this process we were able to find some models that were reasonably successful at picking Formula One podiums. Many of our models had prediction accuracies close to 90% and the best model predicted a podium correctly almost 93% of the time. These metrics mark a significant improvement over our baseline of 85%. For some reason our models were much better at predicting podiums after 2005 than they were for the before data. This may be because certain rule changes after 2005 made some variables more significant when determining podiums. However, because of issues with data processing and downstream model execution errors we were unable to include some of the variables we were interested in. Those include the driver and track which are likely incredibly important for predicting a potential podium. That being said, we were able to find the most important numeric predictors of success. These include points before the race and starting position. We hope that this analysis can shed some light on what makes a successful Formula One race going forward.

Appendix:

RMD and Python files included in the submission