Elliott Bawaan

Using Data to Find the Ideal TDF Winner

For this project, I chose to use a dataset that compiled various information about every Tour De France winner up until 2023. This dataset was put into three files that included different information. The source was from a Kaggle dataset. Some of the information included the physical attributes of each winner, such as age, weight, height, and BMI. As well, the first file includes the length of each of the tours. Additionally, it included the rider type based on PPS. The second file included data on the riders' pre-tour racing statistics. Including columns such as total day races completed, pre-tour stage wins, and pre-tour GC wins. The third and final file had more racing statistics, such as grand tour wins and world championships won.

My goal with this dataset was to try to find which attributes played the heaviest role in building the ideal Tour winner. To do this, I started by doing some visual exploration of the dataset, creating scatter plots and basic plots that examined the trends of the physical attributes of winners like BMI, height, weight, and age. The scatterplots for these four physical features revealed a decent picture for ideal ranges for these categories to succeed in the Tour. For age, weight, and height, I did not find any trends over the years, but I did find what seemed like the ideal ranges for each of the attributes. For age, it seemed that age 26 to 30 was the sweet spot for a winner, while for weight, the ideal weight was between 60 kilograms and 75 kilograms. However, the weight of winners seemed to be trending slightly downwards recently. In the height scatterplot, similarly, there was a slight trend to taller riders over the years, with the ideal range being between 1.75 meters and 1.80 meters. I also looked at some of the external factors, like the evolution of the length of the tour, and compared that to the BMI of the winner. This scatterplot was especially interesting as it seemed to show that as tour length decreased, so did the BMI of the winner. On its face, it suggested an interesting idea that lower BMI may be better for lower distances, which on its face looks counterintuitive, as the longer the tours, the more they may be considered “harder”. But the reason for this is most likely due to the confounding factor that both the winner's BMI and the tour's length have decreased over time. I also created a histogram that examined the number of day races completed before the tour. From this histogram, it confirmed my prediction that most winners were more focused on stage racing and the tour over the one-day events, as most of the winners had less than 10 completed day races. I also created a histogram that looked at the frequency winners from each of the three PPS rider types: sprinter,

climber, and time-trial. This histogram was interesting, as while it was spread pretty evenly, the most winningest rider type was sprinter, while the lowest was climber, which goes against the idea that climbers are the most fit to win the TDF. But I think this just shows that labelling riders into types based on power metrics while ignoring other quantitative and qualitative factors is flawed. The winners of the tour are often well-rounded riders who have the ability to both climb exceptionally well while also having a punchy sprint. For example, both Greg Lemond and Lance Armstrong are listed as time trial rider types when they were both elite climbers who held KOMs for many infamous climbs. Even more, this classification is especially skewed when considering the inaccuracy of power meters up until the last decade.

After this visual exploration, I decided to focus my predictive models on the physical features as they were the most accurate for each winner going back to 1903, and they also seemed the best variables to be able to change, as they are continuous, to find how a rider will perform in a tour. To make my models even better, I added to my dataset with synthetic non-winner riders that followed the trends of the real dataset. This nearly tripled the dataset to enable more realistic predictive models to be run. I decided to run four models and an additional baseline model. The models I ran were linear regression, random forest, K-nearest neighbor, and decision tree. For the baseline, I just used MSE to predict the mean. In each of the four models, I also included the importance that each of the four different variables played in each model.

For the baseline model, I just used MSE to predict the mean. This resulted in a score of .1875.

Starting with the linear regression model, I built it using a pretty straightforward method I prepped the data, then split it into train and test. Then I used a pipeline to optimize the model and fit it. Finally, I ran the model and got an MSE of .1613. So this linear regression model performed slightly better than the baseline model at predicting a winner based on the variables. I then found the importance of each variable using permutation importance. From this digging deeper into the model, the ranking of the importance of each variable was height with a score of .465, followed by BMI at .367, then weight at .273, and finally age at .0151. This suggests that in this model, for the ideal Tour De France winner, height, BMI, and weight play a much bigger role than age. This seems to also be supported by the visualizations I created previously on the actual winner dataset, as there was a pretty solid range of ages that won the tour based on the scatterplot. To summarize the linear regression model did an okay job when predicting a winner

for the tour using the four physical features performing slightly better than the baseline MSE model.

Moving on to my second model, I built a random forest model. To do this, I followed similar steps to the linear regression model, prepping the data and splitting it into train and test. But this time, in my pipeline stage, I used a random forest regressor instead of a linear regressor. I then fit the model and ran it on the test data. This resulted in a score of .0860. Much better than the previous two models, getting a score that was nearly half the MSE of the linear regression model. I then repeated the method to find the importance of each variable to the model. For the random forest model, the rankings were BMI, which was by far the most important with a score of .158, then height with a score of .063, then weight at .0098, and finally age at .000727. This means that BMI was by far the most important factor in the random forest model, and the other three were of relatively little use, especially age, similar to the linear regression model. In short the random forest model was very effective at predicting the winner based on the four variables and leaned heavily on the BMI to get these predictions.

Next was the K-nearest neighbor model. Again, I prepped the data and split it. But then I transformed the data and subsequently created a pipeline for the model. Then I had to do some KNN-specific steps, including defining a parameter grid and cross-validating this grid. After that, I found the best parameter and estimator and finally ran the model. For the test data MSE I got a score of .128. This was a respectable score, performing better than the baseline and linear regression model, but still trailed the accuracy of the random forest model. As far as the feature importance of each feature, the most important was weight with a score of .51, then height at .34, then BMI at .26, and finally age at .15. Once again, like the previous two models, age was the least important variable in the model's predictions. But this model had a much tighter range of scores compared to the previous two, so while it did place a heavy weight on it, it still used the other variables with some level of importance to the predictions. In a nutshell, the KNN model performed well compared to the baseline and linear regression models, but fell behind the strong random forest model.

The final model I built was a decision tree model. This model involved more work to build than the previous models. I split the data and scaled it, but then I had to tune the depth and plot of the train and test MSE. On top of that, I had to build the decision tree itself. Finally, I was able to run the model. This resulted in a test MSE of .166. So this was the worst of the four

models, only beating the baseline model. It performed slightly worse than the linear regression model, but significantly worse than the random forest and KNN models. On the variable importance part of this model, it was the tightest range with weight being the most important with a score of .41, then height at .24, then BMI at .17, and finally, like all the other models, age at .13. This being the tightest range is interesting as it was using a more even share of each variable to make its predictions, but this seemed to come at a detriment to the accuracy of the predictions.

To wrap up the models I built, I built a total of four with a baseline. The two worst performing were the decision tree and the linear regression model, which had MSEs of greater than .16. Then came the KNN model with an MSE of .12. Finally, the best performing model by far was the random forest model with an MSE of .086. Each model had varying ranges of the importance of each variable, but every one valued age as the least important.

After finding my best model, I decided to make a simple web application using Streamlit that allows the user to change each of the four variables, giving them a rider score out of 100 for the ability to win the Tour. This application was built on the random forest model, as it was by far the most accurate model.

Reflecting on this project, I was happy that I found a pretty good model at predicting the winner of a tour based on the four physical characteristics of age, weight, height, and BMI. But there are some things that I would hope to have for even further analysis. First would be to be able to have more data on the composition of each tour, past just the length. Things like the number of meters climbing, the gradients of the bigger climbs, and even more info on the composition of the stages ie, the number of flat/sprint stages, the number of mountain stages, and then the number of time trials. This would enhance my prediction of how certain riders would perform in a certain tour based on their physical characteristics, as it would add more realistic layers of when certain riders may perform better in a given year, with the unique tour route. On top of this optimization of each year's tour route, I would also like to use my app to predict the performance of each rider in this year's tour, and the past few tours, given all these factors. Applying the app to the actual startlist and seeing how it actually performs when it predicts each rider's relative performance to each other would be very interesting.

While the Tour De France has a near infinite number of factors and variables that determine its eventual winner, using past winner data, I was able to build an application on a

random forest model that isolated four physical features and predicts the rider's expected performance in an average tour. Obviously, this model could be optimized even further, taking into account more variables, but it provides a good performance prediction based on these physical features.