**EY Kaggle competition**

**Description**

The data received does not come complete with all the variables required to forecast the race winners. For example, cadence is not a measure that’s always captured and this impacts the result calculations. We are challenging participants to predict the following performance indicators for each cyclist:

* Speed
* Power
* Cadence including any missing data in this field

Importantly, we want to understand the relationship between each of the variables (refer to the ‘data’ tab to see the variables), and its effect or influence on an individual’s overall performance.

Keep in mind that, as the data provided is directly from devices such as activity trackers and mobile apps, there are sometimes errors in the information that’s captured.

Competition rules:

* Evaluating who wins: the total rating between the three categories
* Team effort: We are only allowing one person per submission

**Context**

The professional sport of cycling has lacked sufficient access to real time data that provides insight into the performance of the riders. Individually, this effects the rider’s training and performance in the lead up to a race. As a sport, fans are lacking visibility of a rider’s true performance and output.

Partnering with Strava velon.cc and Vodafone, EYC3 used network data from devices installed on the cyclists and their bikes to provide real-time information on the performance of the individual rider. We built a big data, real-time streaming architecture platform that allowed the processing of varied data sources to provide performance of the cyclist; and visualised to give the rider, team and the fan a new perspective of the competitors’ performance.

Results of each individual rider were pooled together to create a virtual competition that took riders along an identified road route. In the past, we have used Tour De France and Rio 2016 Olympics road race as the backdrop for the simulated race. Tracking the cyclists’ performance in real-time, and applying advanced algorithms on the data helped provide insight into a rider’s calculated relative effort, and their power output, to help predict the performance of each rider to determine who will take out crucial race stages and become the ultimate winner. For more information see: <https://www.youtube.com/watch?v=3_AazV-gjm0>

**Metric**

The error score we will use to evaluate the different models results is the sum of normalised RMSE (Root Mean Square Error) of all three sub-challenges.

In other words, we will calculate the square root of the average of the squared differences between the predicted and the actual values for each category seperately, and then sum it up. This sum will be weighted by the root mean of the squared predicted value for each sample.

The data is seperated into 2 sets:

* Training (70%) contains all columns
* Test (30%) with missing speed (10%), or watts (10%), or cadence (10%)

To submit the results, the parcitipants should send through their predictions in one unique file, in the following format:

* Fist line is for the header
* Use ";" as separator
* Two Columns:
  + Id (same Id as in the Test file
  + Your prediction (for speed, watts ou cadence depending of the missing column

The winner will be determined by having the lowest total error score defined.

**Julien Comment**

\*\*\*HOW PERFS ARE WORKED OUT\*\*\*  
  
Hi all,  
  
some of you had questions about how the perfs are computed. So here is how:  
  
1) For each outcome to predict Speed, Power and Cadence, the plateform computes a subscore as the ratio between the RMSE of the submitted prediction on the 15,000 relevant observations and the standard deviation of the outcome on these samples. Doing so it shows typically how your model shrinks the RMSE compared to the 'model zero' where the prediction is just the average value of the outcome for all the 15,000 samples.  
2) The perf you see on the leaderboard is the arithmetic mean of these 3 subscores  
  
Example : 40% perf can be 1/3\*(10% for Speed + 50% for Power + 60% for Cadence)  
  
Hope this helps,  
  
Julien