The break away in cycling – What makes a breakaway win possible

By: Renan Peneluppi
A capstone project for BrainStation Data Science diploma program

1 – The Problem Statement

A break away in cycling is attempt for one rider, or a small group of riders, to gain distance from the main peloton in order to win a race. In many situations the break away is also a strategy to benefit a team member or to gain some tv broadcast time and make sponsors happy.

In the past the attributes that make a breakaway possible were always approached from the fitness and medical perspective, by measuring max Vo2 from athletes that perform well in these situations, or the power output decay over time from a rider on a small group compared to riders in big groups. As reliable as these are, they approach the training and fitness capacities of individual athletes, rather then race events that may be influenced by strategy.

The idea for this project came to me as I used to ride competitively and have always been a fan of the sport. To me some the most memorable stages in cycling are those where I got to cheer for the underdogs, or the one guy who managed to open a gap and by the end of the race gives everything to make it to the finish line while at every kilometers the gap is smaller and smaller.

From my knowledge and experience of the sport I aimed to prove that besides a few athletic characteristics a profile of a race and specific race events can also influence the outcome of a breakaway.

The goal of this data science approach is to evaluate race strategy based on race characteristics and rider profiles to determine what kind of race and race situation benefits a break away win.

2 – Getting the data

Cycling is a sport currently collecting tons of data, however none of this data is collectively available, as professional riders will usually keep their most valuable data within the teams. In addition, most of the data available is to race results, and not specific in race events. To go around that the collected for this project is for all the Tour de France race stages from 1903 to 2019, except Time trial stages and canceled events.

In addition, every rider that won a stage but later had a result canceled due to doping was dropped from the final data set.

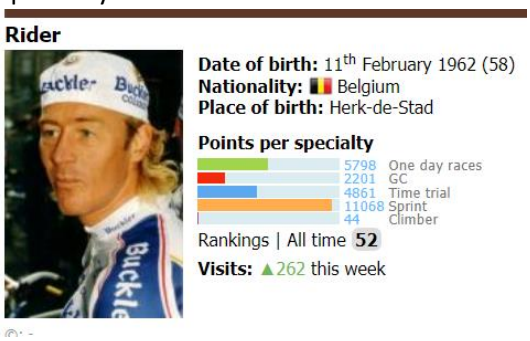
The technique used was web scraping with python pandas and beautiful soup packages, from the website https://www.procyclingstats.com/ for race results and riders information, looping thru every stage and every rider.

All scrapped tables and data were later combined into a full data set for analysis containing over 2600 unique rider details, almost 300 thousand rows and 16 features. This method was effective in combining scattered data, but also presented an expected challenge in data cleaning and normalization, where most definitely most time was spent.

3 – The resulting Dataset

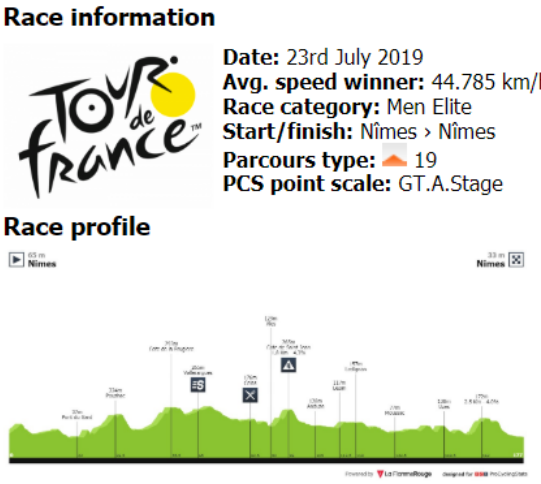
The resulting data contained data for rider profile and race scores as described below.

The Rider data is based on a point system from the source website, where more points are awarded given a rider speciality:



Rider data	Description	Data Type
R_TT	Time Trial speciality	Int
R_onedayrace	One Day race speciality	Int
R_Climb	Climbing speciality	Int
R_GC	Multiple day race speciality	Int
R_Height	Rider height in centimeters	Int
R_Weight	Rider weight in kilograms	Int
R_Sprint	Win in a sprint speciality (Not in a breakaway)	Int
Age	Rider age on race day	Int

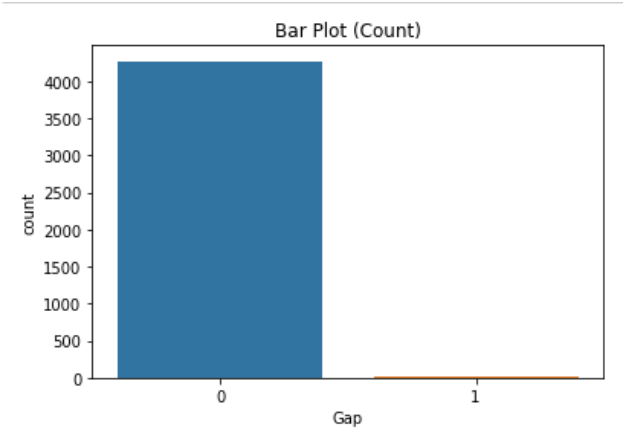
In addition, the data also has the race profile score and additional information, each race is a stage in the Tour de France:

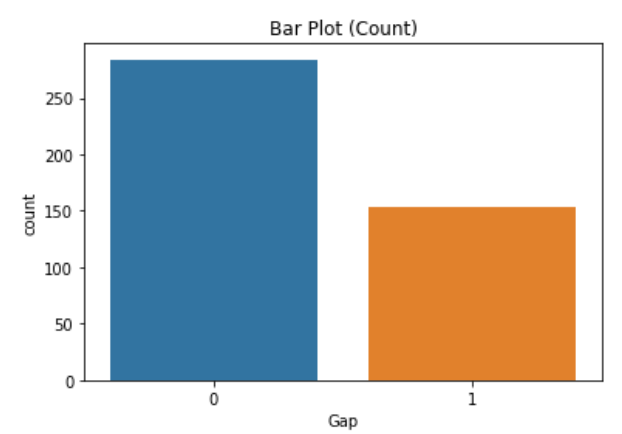


Stage Data	Description	Data type
Ditance	Total distance of the race stage in Km	Float
Rider	Rider Name	Object
Rnk	Rider finishing position	Object
Stage	Number of stage for given year	Object
StageScore	Difficulty of the stage	Int
Team	Name of rider team	Object
Time	Total time to complete the stage	TimeDelta
Winning Avg_Speed	Average speed for the stage winner in Km/h	Float
Year	Year of race	Int
GAP	If victory was with at least 5 second gap to other riders. (A breakaway)	Boolean

A note on Stage Score: The stage sore is calculated by considering the total distance in a stage, the number and difficulty of the mountains (if any), and how close to finish line the mountains are. A lower score represents a flat stage with average distance, while a higher score represents a stage with more mountains, the higher the score the more difficult and closer to the mountains are to finish line. This feature is important in representing numerically what kind of race each stage is, an important factor in race strategy and to combine with riders' strengths.

The final data set turned out to be highly imbalance, as expected when dealing with something that almost never happens, to deal with that I chose, in the end, to compare only the winners of each race:





4 – Data Exploration and Modeling

With such imbalanced data I had to select models that deal better with this kind of data, also I used SMOTE package for up sampling the minority class proportion on the training data, and from that tested a few modelling possibilities: The first attempt was to fit a random forest classifier, considering all race finishers, that turned out as very expensive model for this much data, and resulting in a high precision for majority class but not as high for the minority class. In fact, it would easily predict false negatives for the actual positive cases

The second attempt was to run Logistic Regression models and again Random Forest Classifiers on the same data, but only for the winners of every stage. That demonstrated to not only run much faster but also to have more reliable results. The Logistic Regression model ran fast and had an accuracy of 64%, despite lower accuracy score, it did much better with minority class, correctly predicting it 43% on the test data, versus 30% on the previous expensive Random Forest classifier.

Lastly as I wanted to be able to check data for race events, so I decided to add some more features to the data set. By watching races for 2018 I was able to add, for the final 50km, if there was a break away happening, the distance to the main peloton, and how many riders in it.

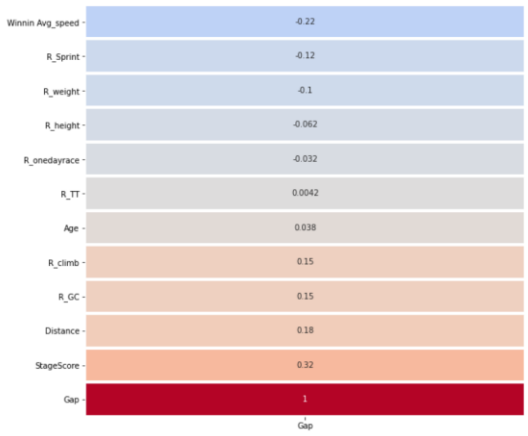
By doing that, even with less data, and using results for all riders, the models performed much better, with precision on the minority class of 66%, while still taking all riders, including no winners, into consideration.

The issue with this is that to collect race event data the process is basically a manual input from long hours watching the races, as this kind of data has never been collected this way. To collect enough data to use it I will need a lot more time in the data collection.

In addition, for a useful model I'd have to run models for all intervals used, so if I was looking at how a race was developing with 40km to the finish line, my model can only take the events up to the last 40km into consideration for the training data. The closer to finish line more features can be considered and more accurate the model is. This would also mean that an API would have to run different models given the km to finish line. I did not have enough time to write this function in a proper reliable way and that is something I intend on doing before making my website public.

```
Confusion matrix of Random Forest
     pred_neg  pred_pos
neg      1062         3
pos         0         6
```

Finally I concluded that, using the original data source, I could rely on key features, such as race profile score, distance and riders that are good climbers to determine the chance of success in a breakaway, but only comparing the winners of each stage and dropping riders that did not win. Despite getting better results I still think it's a waste of data to ignore other riders who did not win stages.



	precision	recall	f1-score	support
0	0.67	0.80	0.73	41
1	0.53	0.36	0.43	25
accuracy			0.64	66
macro avg	0.60	0.58	0.58	66
weighted avg	0.62	0.64	0.62	66

5 – Conclusion and moving forward

From the 3 different approaches I would be more confident with race event data for the last 50km to return much more reliable models. Given the data I had available right now, comparing only the stage winners seems to be the best solution for accuracy.

I can already conclude that race stages with more mountains and higher difficulty will benefit a breakaway with riders with higher climbing skills, or the ones who specialize in on day races. While good sprinters tend to not attempt a breakaway.

The main challenge for this project was collecting, cleaning and normalizing the data, leaving very little time to improve my models. Fitting more models and evaluating different parameters is something I will need to work on before setting the API as public. In addition, to run a confident model I'd prefer to collect more data from race events, I'm confident that once I manage to collect such data, I will have a reliable model for race strategy.

In conclusion, race and rider characteristics do tell a lot, but not enough to properly predict highly imbalanced data. The key race events that I suggested will make the accuracy of the model in the minority class much better but will also be very hard to collect. I did not find any work that took this approach towards the race strategy in the past, I actually plan on collecting data for at least the past 5 years and improving this work with that.