# Unlocking the Tour de France
# Predictive Analysis of Rider Participation Using Machine Learning

Jakob Hutter
Supervisor: Assistant Professor Petra Kralj Novak
Central European University
Vienna, Austria
email:hutter_jakob@student.ceu.edu

## ABSTRACT

This report describes the method and findings of analyzing a web-scraped dataset about pro cyclists including season performance data, race attendance, team affiliation, and personal data. The goal is to predict if a rider will participate in the Tour de France. The report finds, that despite extensive data preprocessing such as one-hot encoding and tuning of different models insufficient prediction results are achieved. This raises the call for further data scraping and time series analysis with more advanced Machine Learning Tools. For the detailed code visit the connected Github Repository.

## 1 INTRODUCTION

In the intricate world of professional cycling, Data Mining has emerged as an essential tool for teams eager to amplify their performance across various dimensions. This project delves into the transformative potential of machine learning in cycling, specifically aiming to predict male rider participation in the paramount 21-day long Tour de France (Male and Female Professionals' Data is not comparable since their racing calendar and race types vary).

Cycling teams harness Data Mining techniques to optimize their strategies, ranging from forecasting rival riders' involvement in specific races to refining overall team tactics. Among these races, the Tour de France stands out as the pinnacle, attracting elite cyclists globally. Acknowledging the Tour's paramount importance, this study investigates whether publicly available data, predominantly sourced from race results, can effectively forecast the participation of riders in this prestigious event.

In contrast to prior analyses, the dataset scrutinized in this project is entirely novel. Through meticulous web scraping, a comprehensive dataset has been curated, featuring variables such as team affiliation, year, rider name, and more. The primary goal is to devise a methodology that utilizes this data to predict Tour de France participants. Additionally, the study seeks to uncover the impact of factors like team affiliation, year, and individual rider attributes on the likelihood of securing a coveted spot in this illustrious race. By the project's conclusion, we aim to provide valuable insights that have the potential to reshape how cycling teams approach rider selection for the Tour de France.

## 2 DATA

The dataset was meticulously gathered through web scraping from the reputable source https://www.procyclingstats.com/ on the 12th of November 2023. The comprehensive data collection process encompassed extracting information for every rider affiliated with World Tour teams (best Teams per season, relegation system similar to premier league) during the expansive timeframe spanning from 2010 to 2023. This involved capturing many details, including team affiliations, race results, and individual rider attributes. The web scraping initiative aimed to compile a rich dataset that serves as the foundation for our predictive analysis of rider participation in the Tour de France.

### 2.1 Data description & Understanding

For simplicity reasons, Tour de France will be referred to as *tdf*. Each column describes a rider's data in a specific region; the data columns can be classified into four categories.

#### *2.1.1 Attributes & Categories.*

- **Team & Season**
  stating the racing team the rider rode for that year (string - categorical), and the season the race data is referring to (int - year, timeseries)
  These variables are important as the each team can only send 8 riders to the tdf. Season can be interesting when considering a time-series application.
- **Rider Infos**
  such as height, weight and age that season (float, float, int)
  Age and weight could be a reason for exclusion of tdf attendance such as "to old" or "to heavy". Height is a constant, that might influence wether a rider rides for a sprint or gc team.
- **Race Attendance**
  stating how many race days till tdf where attended (int), and if on of the other *Grand Tours* (Vuelta Espana, Giro Italia) where attended as well in this season (True, False for both events).
  Normally if a rider attends tdf he will only ride a limited number of days before the event. Since all three grand tours are bodily highly demanding most riders only attend one of the three tours.
- **Performance Data**
  Measures of official cycling federation points (uci) and elo sysmte points (pcs). 3 Different measures 1. till tdf 2. last season 3. this season

#### *2.1.2 Target Variable.*

*tdf attended*
True if rider attended Tour de France the column season, False if he did not.

#### *2.1.3 Missing Values.*

Missing values only occurred for the variables height, weight, and age. All missing values were filled with average values.

## 2.2 Data preprocessing

This section describes the step taken in data manipulation and preprocessing as well as a brief description on the most important implementations.

*2.2.1 Data Preprocessing Steps.*

(1) **Transformation of Age:** Converted age to a relative value within each cycling season.
(2) **Handling Team Name Variations:** Consolidated team name alterations over 13 years to the most recent name for consistency.
(3) **Handling Missing Values:** Addressed missing values in weight, age, and height by filling them with respective average values.
(4) **Rider Identification and Column Dropping:** Identified riders through indices, dropped unnecessary columns (e.g., rider names, PCS points, UCI points) for clarity.
(5) **Categorical Data Encoding:** Employed one-hot encoding for team names to represent them as binary variables.
(6) **Normalization of Non-Boolean Columns:** Scaled non-boolean columns to [0, 1] for improved model performance.
(7) **Season Column Handling:** Used the season column for dataset splitting (pre-2023 for training, post-2023 for testing) and dropped it post-split.

*2.2.2 Implementation Details.*

The data preprocessing steps utilized foremost methods from web-scraping module *soup* while collecting, secondly *pandas* methods for alterations of the table and self-built functions for specific alterations of one-hot encoding to splitting. Due to high computing effort, specific tasks, such as team category manipulation, were separated into distinct scripts for enhanced efficiency ((1),(2),(3) in *data_manipulation.py*; (4),(5),(6),(7) in file *data_preprocessing.py*)

Enabling a more in-depth exploration of how team affiliation could impact prediction models, we employed a built-in function called *local_variables*. This function accepts a boolean parameter and returns the X and Y test and training sets, allowing to choose whether or not to include team affiliation information in the datasets.

# 3 MACHINE LEARNING METHODS USED

This chapter delves into the rationale behind the selection of Decision Tree and Random Forest classification algorithms, despite the comparable efficacy of regression models the experiment scenarios.

The dataset encompasses a diverse set of features, including numerical variables such as age, height, weight, pcs points till tdf, uci points till tdf, race days till tdf, last season pcs points, and last season uci points. Additionally, binary indicators for attendance in prestigious cycling events, namely vuelta and giro, as well as categorical variables denoting team affiliations, add complexity to the predictive modeling task. The inherent nature of the problem, characterized by the need to predict categorical outcomes related to cycling participation, aligns with the versatility offered by decision tree-based algorithms. Decision Trees, with their hierarchical structure of decision nodes, are adept at handling both numerical and categorical features, making them suitable for the diverse data types present in the dataset.

Furthermore, the choice of Random Forest, an ensemble method built upon Decision Trees, is justified by its inherent ability to mitigate overfitting and enhance model generalization as an add to the decision tree.

While regression models could yield comparable results after tuning, the interpretability and inherent feature importance analysis provided by Decision Trees and Random Forest make them particularly appealing for a domain where extracting actionable insights is as crucial as predictive accuracy.

## 3.1 Parameters

*3.1.1 Grid Search Parameters.*

(1) **CV:** Number of cross-validation sets, 5 seems reasonable.
(2) **Score:** F1 score for evaluation, see Experimentation.

*3.1.2 Decision Tree Parameters.*

- **Quality of Split**
  (1) **Entropy:** Used for information gain in the splitting process. Makes it more efficient, as linear regression showed some attributes are very strong indicators for classification.
- **Splitter Best Split:** Chooses the best possible split for nodes. We don't want to find a random but optimal split
- **Max Depth** maximum of 15 to avoid overfitting.
- **Min Samples Split:** Minimum number of samples required to split an internal node during tree construction. Since the dataset is quite large, a larger number should be considered, I added 50-160 range.
- **Min Samples Leaf:** Minimum number of samples required to be at a leaf node to reduce overfitting. This would not be needed necessarily as the dataset is very large and has float values.
- **Max Features:** Number of features to consider when looking for the best split. None, Sqrt, log2, 3 given as options, None would be sufficient.
- **Random State:** Seed value to produce the same results in each run. Random int such as 7 or no seed.

*3.1.3 Random Forrest Parameters.*

(1) **n_estimators:** Number of trees. Good range: 10, 50, 100.

rest is similar to Decision Tree parameters

## 3.2 Brief description of the evaluation criteria

see *Classification Evaluation*

## 3.3 Results

The tuned Decision Tree has an F1 score of 60% for True Positives, while the Random Forrest scores 60%.

# 4 EXPERIMENTS

This section outlines the experimental procedures undertaken to identify a suitable model for hyperparameter tuning and to determine additional data preprocessing measures. Additionally, the exploration encompasses the experimental investigation of evaluation metrics that hold significance for the given dataset.

## 4.1 Iterative Eperminatal Approach

This experiment had an iterative approach, meaning that the models in section *Models Assesd* were reevaluated, examined and altered in correspondence to experiments:

- Inclusion/exclusion of team affiliation encoded columns.

- Data normalization for feature standardization.
- experimenting with evaluation crierias (Recall, Precsion, F1, printing ROC,...; MSE, RMSE,...)

## 4.2 Models Assessed

- Regression Models
  - K-Nearest Neighbors (KNN)
  - Linear Regression
  - Decision Tree
- Classification Models
  - KNN
  - Decision Tree
  - Naive Bayes
- Classification Ensembles
  - Random Forest
  - Ada Boost
- Explored Neural Networks
  - MLP Classifier

## 4.3 Evaluation Criteria

### 4.3.1 *Classification Evaluation.*

The goal of this project is to increase the True Positive predictions as high as possible. As a result, the F1 score for True values was found to be the most accurate measure for evaluation. This comes as both False Positives and False Negatives want to be avoided as every rider wrongfully predicted is a deviation in team strategy.

### 4.3.2 *10 fold cross-validation.*

The models were tested using 10-fold cross-validation to robustly assess their performance across different subsets of the training data, ensuring a reliable estimation of their generalization capabilities and reducing the impact of variability in the data splits.

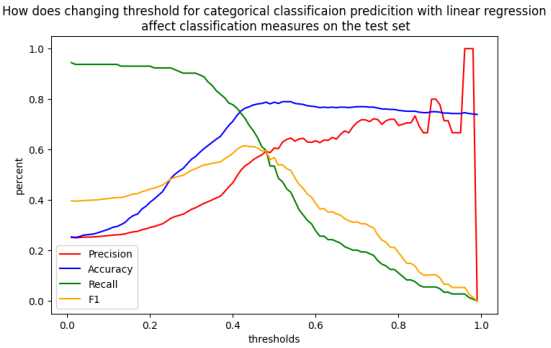
### 4.3.3 *Findings.*

(1) Running models without team affiliation improves model performance, indicating that rider individual data is more important
(2) balanced outcome can well be assessed with F1 score, either

# 5 VISUALIZATION

## 5.1 Finding Classification Threshold

Utilizing linear regression tdf attendance can be classified by prediction. This graph visualizes how different classification measures change in this dataset, and evaluation on the test set.




It is quite interesting to see that there is a tradeof between between precsion and recall.

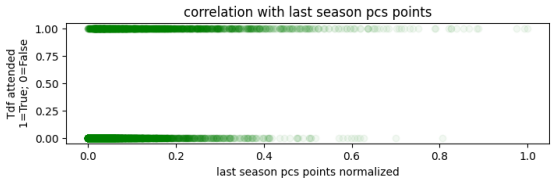
## 5.2 Correlation

First of all using linear regression can be a strong tool to present data, as the coefficients for each normalized attribute, indicate how strong a indicator each is for predicting tdf participation.

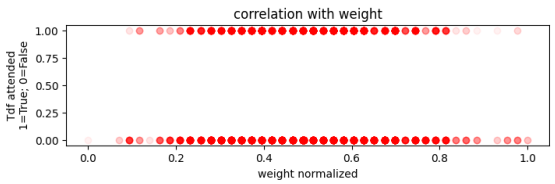
| Feature | Coefficient |
|---|---|
| Last Season PCS Points | 1.3356 |
| Age | 0.4606 |
| UCI Points till TDF | 0.3971 |
| Height | 0.3222 |
| PCS Points till TDF | 0.3206 |
| Giro Attended | -0.2596 |
| Weight | -0.2186 |
| Race Days till TDF | 0.2050 |
| Vuelta Attended | -0.1573 |
| Last Season UCI Points | -0.1214 |

**Table 1: Regression Coefficients for Feature Importance**

Observing that the Elo point system (PCS) of the past significantly influences the prediction of Tour de France attendance underscores a logical correlation. This observation aligns with the inherent nature of the Elo method, which defines an athlete's strength within the network. In contrast, UCI points are solely distributed based on race results. As well as the reason that Grand Tour attendance is mostly decided before the season, being thee reason why "till tdf" indicators are weak. The discerned emphasis on the Elo point system in predicting TDF attendance is consistent with its role in characterizing the athlete's competitive prowess within the broader network, contributing to a more comprehensive prediction model. See figure below.

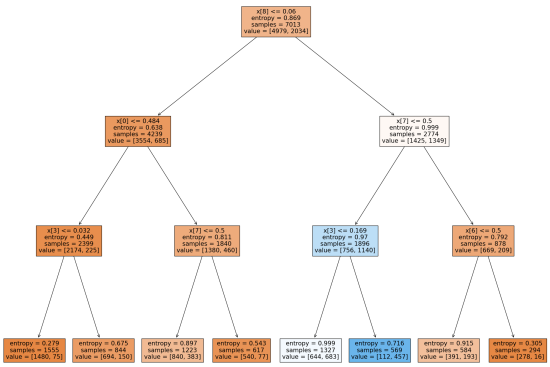



In comparison, the attribute with a comparatively weak predictive influence is weight, as indicated by its correlation graph below.



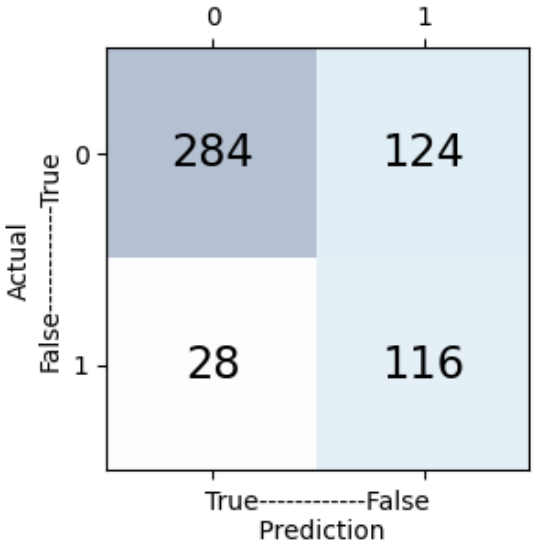
## 5.3 Decision Tree

As an example the first 3 layers of the optimized decision tree



## 5.4 Model Findings

In the visualization depicted is the confusion matrix for the decision tree results with hyperparameter tuning.




# 6 CONCLUSION

This work focused first on harvesting data online that can be useful. Secondly, on finding a model that can accurately predict future outcome. While it was expected that classification performs well, the result of experimentation that regression and Neural Network models perform similarly well, was suprising.

The goal of this project was to provide a prediction tool that accurately predicts which riders will attend next year's tour de France by utilizing publicly available data. With neither precision nor F1 score being higher than 0.7 in any of the models, it can concluded that the goal was not reached. This is the case as depending on which riders are attending, the race style can heavily influx, as riders have different specialties. For example, if more riders with the goal winning in the high mountains come, a team with both good climbers and sprinters, would let their sprinters attend the race, when prediction accuracy measures are as low as here, there it is better to rely on old methods.

This outcome is not unexpected, as the data scraped is very limited, and other more extensive measures can be taken:

(1) Consider Time-Series Analysis, such as understanding how riders performance trajectory, age increase, and team can affect this.
(2) Include more detailed data on specialties ranking, is a rider good in climbing, time trials, one day races. A draft for a scraper that harvests is already built, but would need extra attention for detailed debugging.
(3) There are also rules such as "each team can send 8 riders per year to the tdf" that can have influence prediction positively if included.

Improving the method of predicting riders participation on the Tour de France could potentially used for any other race and utilized by each team at the end of the season to set their own race calendar in a way to potentially maximize their uci points, which are the basis for relegation and therefore related to teams income in the future.

# 7 REFERENCES


- Github Repository
- sklearn documentation
- matplot lib documentation
- pandas documentation
- soup documentation