University of Pisa

DEPARTMENT OF COMPUTER SCIENCE

Data Mining Report

Group 14:

Tommaso Di Vito

Filippo Boni

Saul Urso

ACADEMIC YEAR 2024/2025

Contents

1 Data Understanding

The given files represent professional cycling racers and races. The datasets we work on are: *cyclists.csv* and *races.csv*, both of which are tabular representations of the respective entities.

Specifically, the *cyclists.csv* table represents each cyclist with the following attributes:

- Name
- Birth year
- Weight and height
- Nationality

On the other hand, the *races.csv* table describes the following properties for each race contender:

- Race identifier and name
- Old point system of the race and UCI points
- Date, length, climb altitude, soil type, and average temperature of the race
- Position of the cyclist, their age, time difference from the winner in the race, and their team

We will later on in the Data Cleaning phase discuss the specific columns of the datasets and proceed with the removal of the surplus ones.
In this part of the chapter, we focus on understanding the data without manipulation as required by the project instruction.

1.1 Cyclist Pre-Analysis

1.1.1 Weight and Height Distributions

Analyzing the distribution of weights and heights is key to understanding the physical characteristics of professional cyclists, as these attributes may correlate with performance in different race profiles, such as mountainous or flat terrains. It's also important to recognize that cyclist generations have become progressively more trained over time influencing the data. Figure 1 illustrates that the data closely follows normal distributions, with weight peaks around 67 kg and 72 kg, and height peaks around 185 cm and 178 cm.

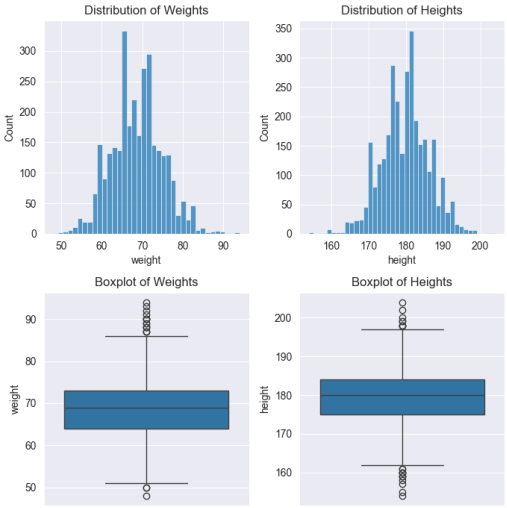


Figure 1: Distribution and Boxplots of Weights and Heights

1.1.2 Height by Nationality

Figure 2 shows the distribution of weights by nationality shows regional differences in cyclists' physics. Such differences may reflect training practices, genetic factors, or selection criteria for professional cycling. The distribution of nationalities highlights the diversity and regional representation of cyclists in the dataset. Italy emerges as the leading country, with over 1,000 cyclists. Belgium and Spain follow closely with similar numbers. We also analyzed birth years across different nationalities: Notably, Italy, France, and Belgium show a strong historical dominance in cycling. In recent years, however, a shift is evident, with younger cyclists from other countries emerging. Meanwhile figure 2 shows how Norwegian racers have an average higher weight compared to other countries' contenders. We can observe also that the average weight of the most common nationalities (derived from the previous analysis) contenders are very similar, showing how most contenders have similar physiques.

Figure 2: Weight Distributions by Nationality

1.1.3 Trend of Weight and Height Over the Years

The evolution of cyclists' weight and height over birth years demonstrates how physical characteristics may have changed over time, potentially influenced by shifts in training, nutrition, or selection standards. Figure 3 displays median and mean trends to capture these dynamics.

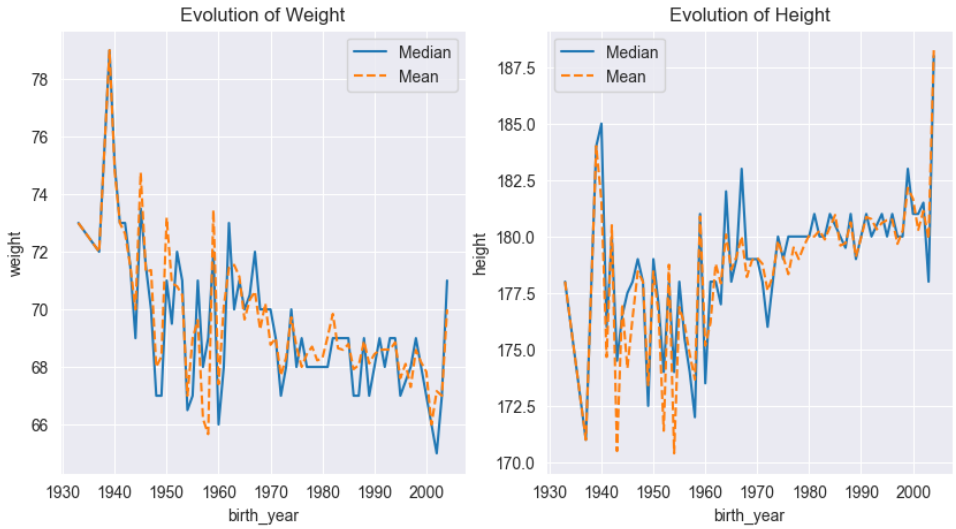


Figure 3: Evolution of Cyclists Weight and Height Over the Years

Figure 3 shows this trend: the average weight of cyclists has decreased, while their average height has increased. Combined with previous analyses of age, these trends support the hypothesis of a shift in nationalities, particularly among younger cyclists. The decrease in weight and the increase in height further highlight the significant changes in training regimens over time.

1.1.4 Correlation Analysis

Understanding correlations between attributes such as height, weight, and birth year can show interdependencies within the dataset. The correlation matrix in Figure 4 shows a high correlation between the height and weight of the cyclists, as we expected.

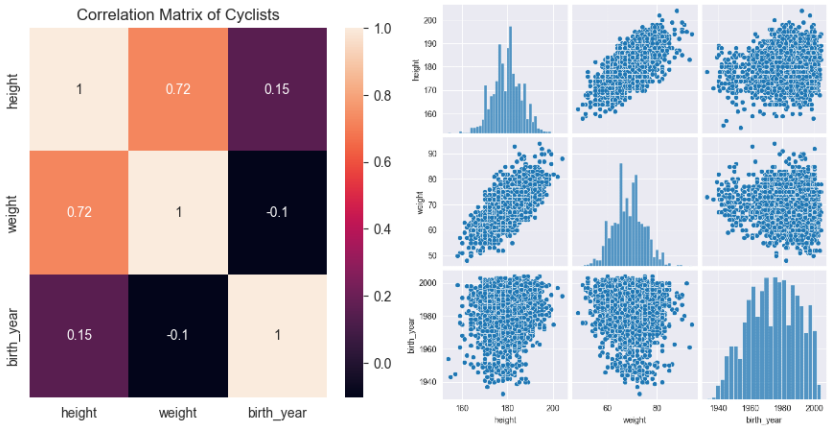


Figure 4: Correlation Matrix and Pairplot of Cyclists' Attributes

1.2 Races Pre-Analysis

An initial assessment of the data in *races.csv* revealed the following: We observed that the *cyclists.csv* table was incomplete, with some cyclists missing key attributes such as height, weight,

and birth year. Additionally, certain cyclists had no recorded participation in any race. Notably, the nationality of the cyclist *Scott Davies* was also missing. **The following discrepancies were found in the dataset and addressed in the next Data transformation/cleaning chapter.**

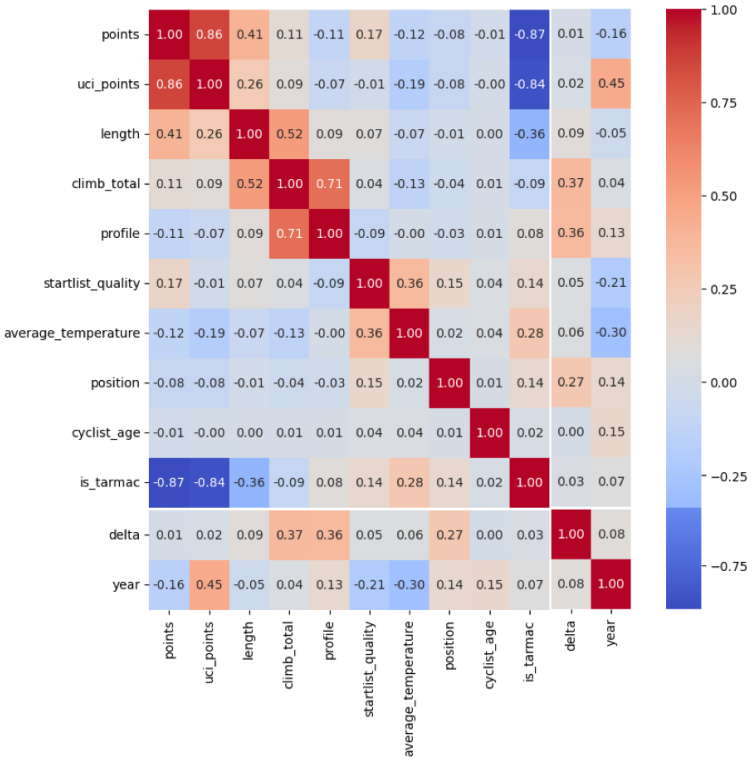
- The columns *is cobbled*, *is gravel* are all set to *false*, while *is tarmac* is always *true*, implying that all races took place on tarmac. Therefore, this categorical attribute was excluded from next chapters' analyses.
- Some races had missing values for *climb_total* and *profile*, and the average temperature data of the races was frequently missing.
- The *delta* attribute, representing the time difference between a cyclist and the race winner, contained incorrect values (e.g., negative times).
- The *UCI point* system replaced an older points system in a certain year. These two systems are highly correlated numerically.
- Missing race points were retrieved through online data scraping.
- Some cyclists appeared multiple times in the same race, occasionally with identical positions and at other times with differing positions.
- Racers with *position* equal to 0 were the first to arrive

1.2.1 Correlation Analysis

From Figure 5, we can observe the correlation between the numerical attributes of the races dataset. The darker colors highlight a strong correlation between *UCI points* and *points*, which aligns with the historical context discussed earlier.

Additionally, there is a moderate correlation of 0.46 between *points* and *length*, suggesting that race length might be one of the factors considered in the calculation of points. There is also a strong correlation between *climb total* and *profile*, indicating that the race profile is heavily influenced by the total climb.

Figure 5: Correlation Matrix of races attribute



2 Data Transformation

In this section, we focus on the data cleaning tasks that addressed the inconsistencies identified in the data understanding chapter as well as newly emerged ones. These tasks retained only meaningful attributes,

resolved inconsistencies in individual values, and, where feasible, corrected missing or erroneous data through web scraping. This chapter shows how we proceeded with novel feature engineering to identify and capture interesting relationships between cyclists and races, how we performed outlier detection, and, a renewed data understanding phase on the improved datasets.

2.1 Cyclists Data Transformation

The following *data-cleaning* steps were taken to address issues in the datasets:

- Confirmed that the `_url` column in *cyclists.csv* is unique and contains no missing values (it is a valid key).
- Verified that all cyclists in *races.csv* exist in *cyclists.csv* (consistency between the datasets).
- Addressed missing *birth_year* values for specific cyclists using trusted sources (*procyclingstats.com*).
- Filled in missing *nationality* data, assigning "Britain" and birth year for cases like *Scott Davies*.
- Removed cyclists with incomplete or invalid data who did not participate in any races.
- Performed dataset merging while identifying and resolving mismatches to ensure consistency.

To address missing values in the *weight* and *height* attributes, we utilized a K-Nearest Neighbors (KNN) imputation method. By considering the values of the 5 closest neighbors, missing data was inferred based on similar cyclists' attributes. This will provide a consistent dataset while preserving underlying relationships between these features.

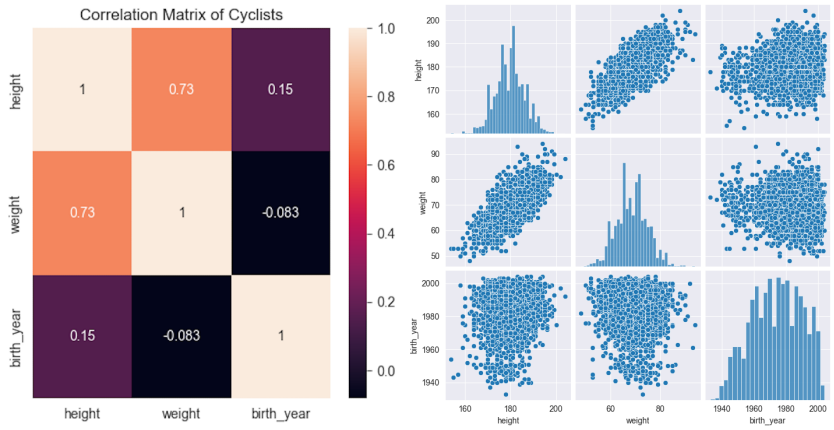


Figure 6: Cyclists attributes after imputation

As shown in the correlation matrix in Fig. 6, the correlation between the *height* and *weight* attributes slightly increased following data cleaning and imputation. This aligns with expectations, as missing or inconsistent values were addressed, leading to more accurate relationships compared to Fig. 4.
For the **Novel Feature Selection**, we introduced a new feature: the **BMI** (*Body Mass Index*) of cyclists. This feature aims to capture the relationship between a cyclist's weight and height, showing physical fitness and potential impact on performance, especially in endurance and climbing-intensive races. The BMI was calculated using the formula:

$$\text{BMI} = \frac{\text{weight (kg)}}{\left(\frac{\text{height (cm)}}{100}\right)^2}$$

This computation was applied to each cyclist, and the resulting values were added as a new column in the *cyclists.csv* dataset.

2.2 Stages Data Transformation

We considered grouping the *races.csv* dataset by stages of each race, as the dataset includes a unique identifier for each race stage. This approach allows us to analyze performance trends and characteristics at a finer granularity, capturing how factors such as terrain, length, and climb difficulty vary across stages. Grouping by stages also shows how cyclists perform in multi-stage events.

We will from now on use this new *Stages* dataset for next chapter analysis as provides a finer level of detail compared to the original *races* dataset

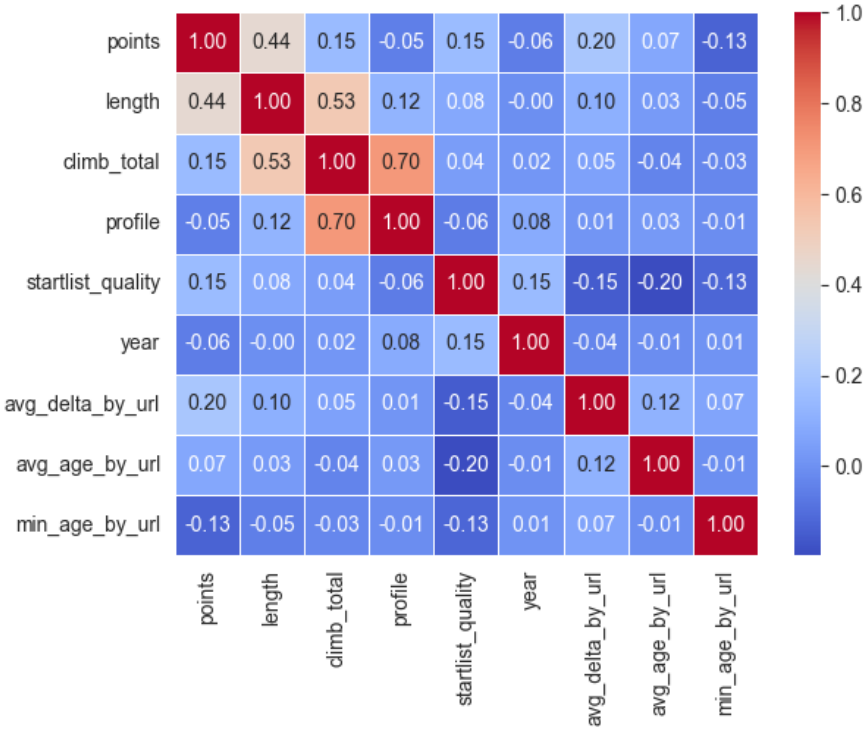


Figure 7: Correlation of Aggregated Attributes on Stages

We grouped race stages and computed attributes like average and minimum age of cyclists, as well as the average and maximum time deltas from winners, we aimed at capturing the dynamics of the races.

From Fig. 7, a correlation exists between *climb_ total* and *profile* (0.70), showing what already seen before that a race's climbing demand strongly shapes its overall profile. Similarly, a moderate positive correlation (0.44) between *length* and *points* shows that longer races contribute to point allocation. The lack of high correlations among the new features (min/max and average) shows the diverse nature of race outcomes.

For the **Novel Feature Selection**, we introduced a new feature: **Season**, derived from the time of year the race occurred (e.g., winter, summer, etc.). This feature provides context regarding environmental and weather conditions, which can significantly impact race dynamics and cyclist performance. Seasonal variations influence factors like endurance, energy expenditure, and even equipment choice.

2.3 Races Data Transformation

2.3.1 Handling Delta Inconsistencies in Races Data

The *delta* attribute in the *races.csv* dataset, which represents the time difference between a cyclist and the race winner, was found to have two primary issues:

- Some *delta* values were negative, which is logically incorrect as a cyclist cannot finish before the winner.
- The *delta* values were recorded at the team level rather than for individual cyclists, leading to inaccurate representations of individual performance.

2.3.2 Distribution Analysis of Delta Values

To better understand the extent of these inconsistencies, we analyzed the distribution of the *delta* values before any corrections. The histogram (Figure 8) shows negative values and anomalous peaks.

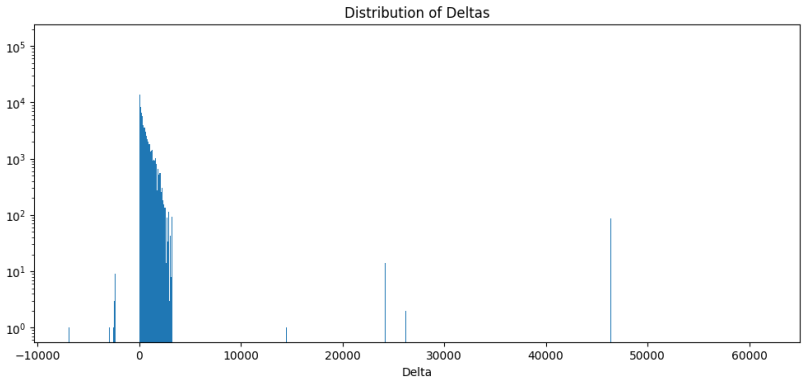


Figure 8: Distribution of *Delta* values before scraping and cleaning.

2.3.3 Web Scraping for Delta Correction

To address these issues, we scraped the correct values on the online database *procyclingstats.com*. The process generated a new dataset, *races_ delta_ scrape*, which provided more accurate *delta* values for individual cyclists. We compared the pre- and post-scraping *delta* values for a subset of problematic race stages, as shown in Figure 9.

2.3.4 Distribution of Deltas After Scraping and Cleaning

After scraping, the distribution of *delta* values improved significantly, showing fewer anomalies. We then proceeded with a final cleaning step to ensure the dataset's consistency producing the results shown in Figure 10.

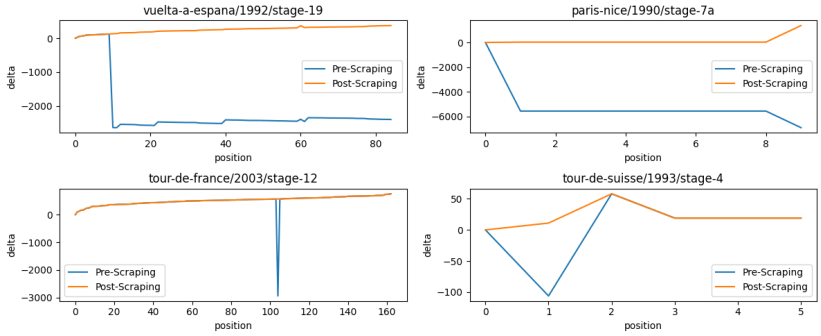


Figure 9: Pre/Post scraping delta value comparasion on sample races

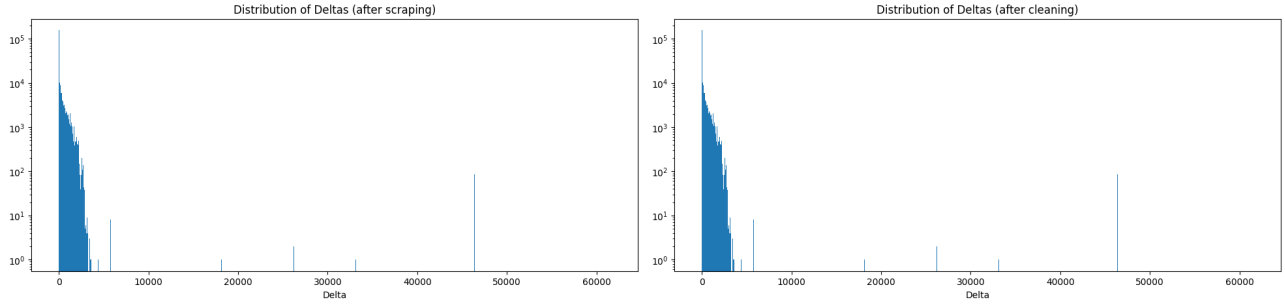


Figure 10: Distribution of *Delta* values after web scraping and Cleaning

Through these transformations, the *delta* values in the races dataset were corrected, ensuring they accurately reflect individual cyclist performance essential for the reliability of subsequent analyses.

2.3.5 Imputation of Missing Profiles Based on *Climb_ Total*

The *profile* attribute, which categorizes races into difficulty levels, was found to have missing values in some race stages. To address this, we exploited the *climb_ total* as it showed a strong correlation with *profile*, making it a suitable basis for imputation.

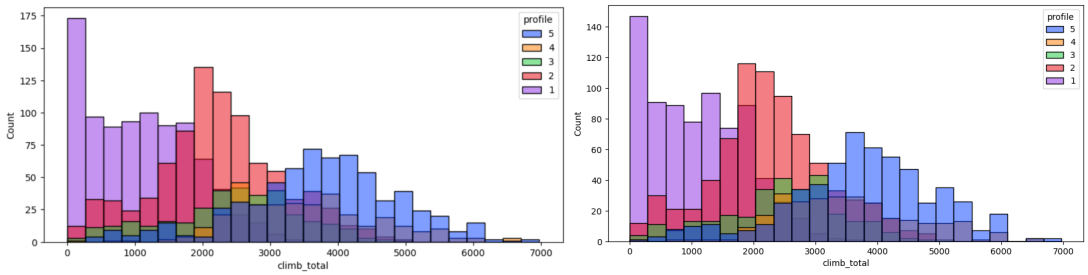


Figure 11: *Climb_ Total* values categorized by *Profile* pre imputation (left) and post- (right).

Imputation We computed the correlation between *profile* and *climb_ total* using Spearman's and Kendall's methods. These confirmed a significant correlation, making *climb_ total* a reliable predictor of *profile*.
We used KNN imputation with one neighbor and a distance-weighted approach to predict missing *profile* values based on the *climb_ total* attribute.
Post-imputation, the distribution of the *climb_ total* attribute across different *profile* categories. In Figure 11, *climb_ total* distribution is showed across imputed and original *profile* levels.

Removal of *Climb_ Total* Given the strong correlation between *climb_ total* and *profile*, the former was removed to reduce redundancy and simplify the dataset.

2.4 Anomaly Detection

Anomaly detection was performed on the two datasets: *Cyclists* and the newly generated *Stages* as in our opinion was easier to handle and more representative of the races at a finer level. We wanted to identify and remove outliers while preserving the core distribution of the data.

Cyclist Dataset Anomalies For the *cyclist* dataset, final assignments were validated based on empirical observations, as alternative methods produced suboptimal results. The preprocessing involved standardizing numerical features such as *height*, *birth_ year*, and *avg_ pos*, followed by anomaly detection using the following techniques:

- **Gaussian Mixture Models (GMM):** AIC and BIC metrics were used to optimize the number of components.
- **Connectivity Approach:** Local Outlier Factor (LOF) with varying k neighbors, evaluated using silhouette scores
- **Support Vector Machines (SVM):** One-class SVM with different kernels (RBF, polynomial, linear) to identify boundary outliers.
- **Isolation Forest:** Decision functions and anomaly scores were used to highlight isolated points.

Outlier removal was conducted through a majority vote across these methods, ensuring robustness.

Stages Dataset Anomalies The same methodology was applied to the *stages* dataset, focusing on attributes such as *year*, *length*, and *startlist_ quality*. Distributions were examined pre- and post-removal to ensure the preservation of data integrity.

While the previous Fine-tuning techniques were attempted, the results were unclear, so we opted for empirical rules to approach the anomaly detection on the datasets.

2.5 Data Understanding with new features

We proceeded with a new data understanding phase on the new seasonal attribute:

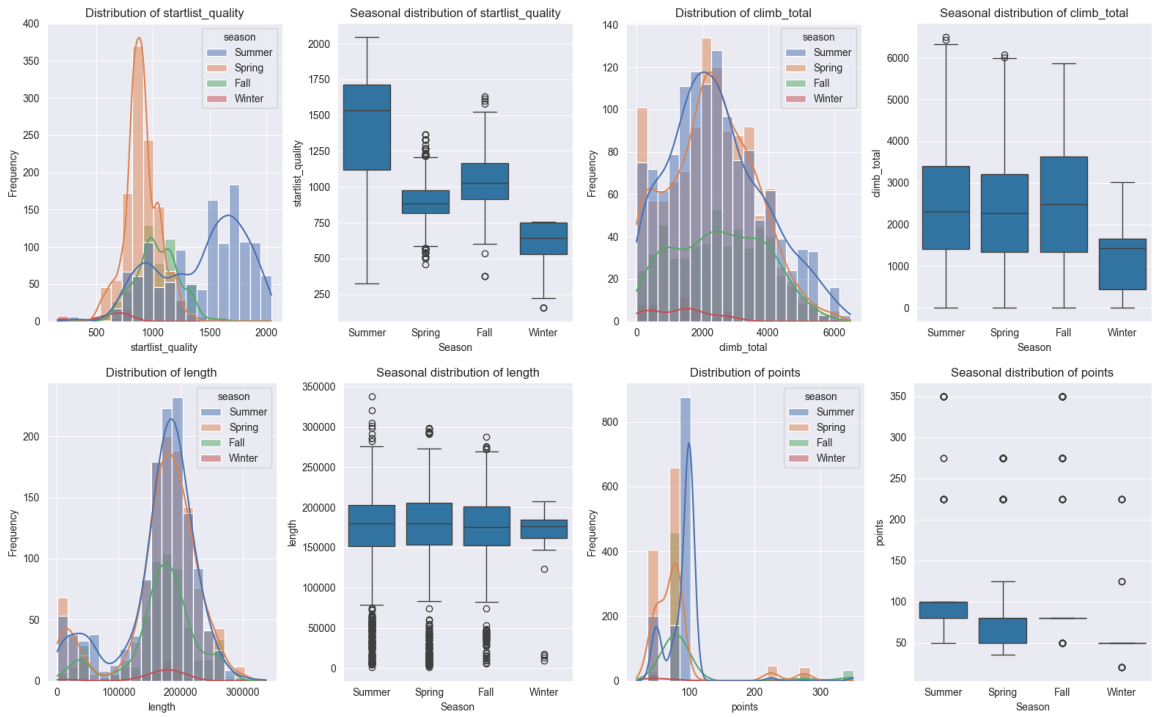


Figure 12: Bar and box plots of Start-list quality, climb, length, and points by season

In figure 12 the seasonal analysis of key features in the dataset is produced. The distributions of `startlist_quality`, `climb_total`, `length`, and `points` show clear seasonal variations. Summer has the highest medians and variability across most metrics, indicating increased activity. Winter exhibits lower medians and reduced variability, suggesting less activity. The boxplots reinforce these findings, showing distinct seasonal patterns.
Seasonal analysis of additional features in the dataset was done. The average age showed higher medians in Summer and Fall, with Fall exhibiting more variability. `Profile` values remain consistent across seasons, with minimal differences. The average delta (time difference between winner and others) has the highest medians and variability in Spring and Summer, implying seasonal impact. The `year` feature shows an increasing trend over time, with Winter containing more recent data.

3 Clustering

We conducted a cluster analysis to identify common patterns among different stages and cyclists. Specifically, for each of these two groups, we tested various types of clustering algorithms:

- K-Means and X-Means for distance-based clustering
- Agglomerative and Spectral for hierarchical clustering
- DBScan and Optics for density-based clustering

We only report the results obtained using advanced outlier detection techniques (e.g., Isolation Forest, Gaussian Mixture, etc.). See the Notebook "Clustering with AD" for more details. Also we specify that we report only the results for the stages table since for cyclists we got no good and interesting results.

3.1 Hyperparameter Tuning Techniques

3.1.1 K-Means and X-Means

- **K-Means**: To find the optimal K, we used the elbow method and refined the selection by evaluating the silhouette score. Specifically, we chose a K value where the number of clusters was neither too large nor too small according to the elbow method, while maximizing the silhouette score.

- **X-Means**: X-means only requires setting the minimum number of clusters and the initial configuration. We used 2 as the minimum number of clusters and we executed the algorithm more times with different starting configurations and we took the most common result.

3.1.2 Agglomerative and Spectral

- **Agglomerative**: We determined the optimal number of clusters using dendrogram visualization and the elbow method for agglomerative clustering using the K-distance linkage plot when it was difficult to understand the results of the dendrogram. By identifying the largest gap, we selected the cut height that yielded the most meaningful clusters. We further fine-tuned the cluster count by evaluating the silhouette score across a range of values, ultimately selecting the configuration that balanced optimal scores and visual clarity.

- **Spectral**: We identified the optimal number of clusters using the eigengap method. After constructing the Laplacian matrix, we located the index with the largest gap between eigenvalues (excluding zeros) and used it as the cluster count. We further fine-tuned the results based on the silhouette score to select a configuration that maximized the score and produced visually interpretable clusters.

3.1.3 DBScan and Optics

- **DBScan**: To find the optimal epsilon, we used the K-Distance plot fine tuning the result looking at silhouette and number of noisy points found. For the Min_Samples parameter, we tested values based on the thumb rule ($2 * \#columns$), the curse of dimensionality rule ($2^{\#columns}$) and the default value (5). We refined the configuration by selecting the interval that maximized the silhouette score.

- **Optics**: We determined the optimal epsilon using the Reachability Distance Plot fine tuning the result looking at silhouette and number of noisy points found. For Min_Samples, we applied the same approach as in DBScan.

Note: Although we could have further fine-tuned the Min_Samples parameter, we decided not to proceed due to the already satisfactory results.

3.2 Stages

3.2.1 Distance-based Results

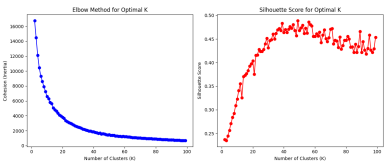


Figure 13: K-means Cohesion vs. Silhouette over different K

Figure 14: Proximity matrix for K-Means (Best K)

For the K-Means algorithm, We identified the optimal range for K using the elbow method and fine-tuned it by examining the silhouette score. Based on this, we can see from figure 13 that 12 clusters is the optimal number. The proximity matrix in figure 14 shows some distinct low-score blocks on the diagonal, suggesting that the clusters are reasonably well-defined, indicating we may be capturing meaningful patterns.

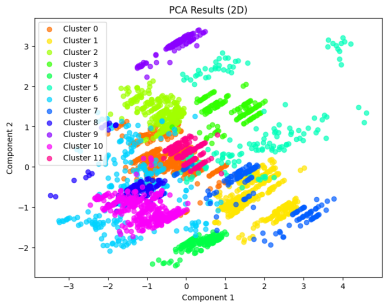


Figure 15: PCA for K-means

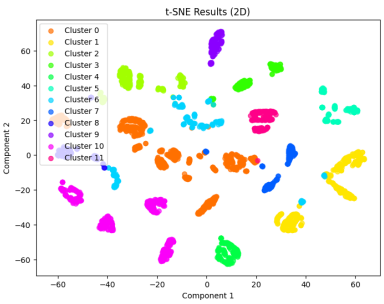


Figure 16: T-SNE for K-means

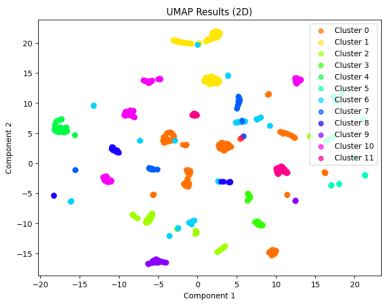


Figure 17: UMAP for K-means

The PCA visualization (figure 15) shows some overlap between clusters, but most of the time, the clusters do not overlap, and the points within the same cluster are close to each other. Using T-SNE and UMAP (figures 16, 17), we observe that the clusters are well-separated, and points within the same cluster are very close. Additionally, the silhouette score is sufficiently high, suggesting we are likely capturing meaningful clusters.

Figure 18: Proximity matrix X-means

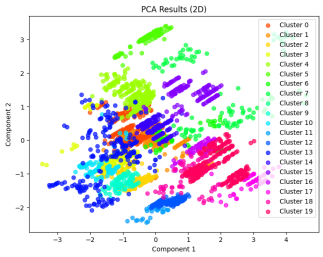


Figure 19: PCA X-means

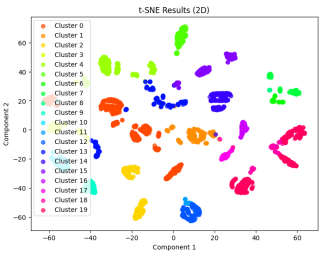


Figure 20: T-SNE X-means

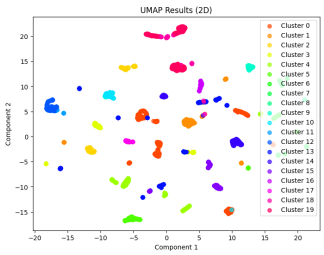


Figure 21: UMAP X-means

We also experimented with X-Means, and the results were very similar to those from K-Means, though X-Means identified more clusters. The proximity matrix (figure 18) is quite similar to the one from K-Means, with distinct low-score blocks on diagonal. From the visualizations, we can see that while PCA (figure 19) shows some cluster overlap, most clusters are well-separated, and points within the same cluster are close to each other. T-SNE and UMAP (figures 20, 21) show better separation of the clusters, suggesting that we are likely capturing the same patterns as with K-Means.

3.2.2 Hierarchical Results

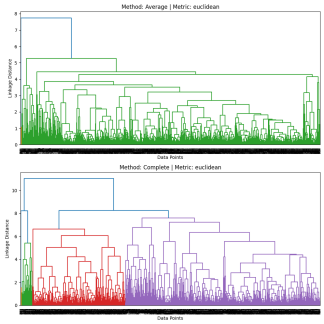


Figure 22: Linkage Results

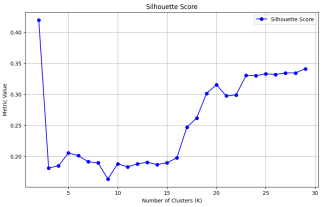


Figure 23: Agglomerative silhouette varying N° Clusters

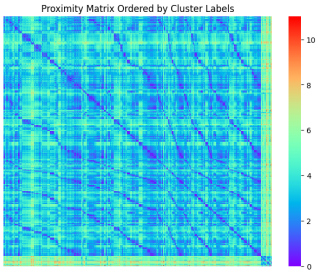


Figure 24: Proximity matrix Agglomerative

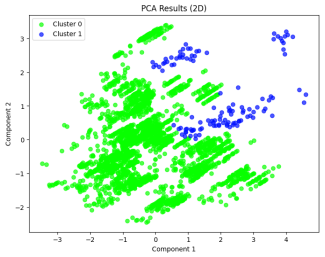


Figure 25: PCA Agglomerative

For the agglomerative clustering algorithm, we used the dendrogram to identify the optimal range for the number of clusters. We can see from figure 22 that the optimal point for the cut is not very clear, so we applied the elbow method for agglomerative clustering and fine-tuned the result using the silhouette score (figure 23) (see the notebook for more plots). Combining these techniques, we determined that the optimal number of clusters is 2. The proximity matrix (figure 24) does not show distinct low-score blocks, and the visualizations using PCA (figure 25) indicate that the two clusters are not perfectly separated, with one cluster being much larger than the other. Thus, we can conclude that we are likely capturing a random pattern.

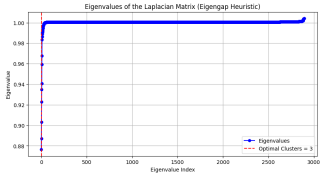


Figure 26: Spectral Eigen Gap

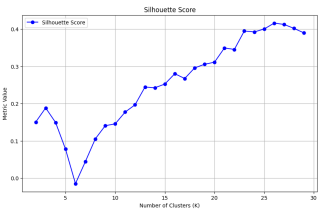


Figure 27: Spectral silhouette varying N° Clusters

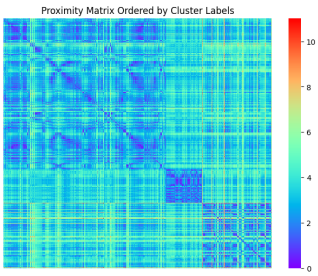


Figure 28: Proximity matrix Spectral

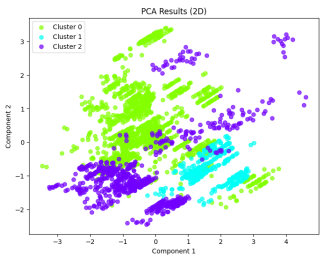


Figure 29: PCA Spectral

For the spectral clustering algorithm, we used the eigen gap method to identify the optimal range for the number of clusters (figure 26), and fine-tuned the value by examining the silhouette score as we varied the number of clusters (figure 27), finding 3 clusters as optimal value. In this case, the proximity matrix (figure 28) does not show distinct low-score blocks, and the visualizations reveal that the clusters frequently overlap, with points within the same cluster appearing far apart from each other. After this analysis we can say that we are capturing some random pattern.

3.2.3 Density-Based Results

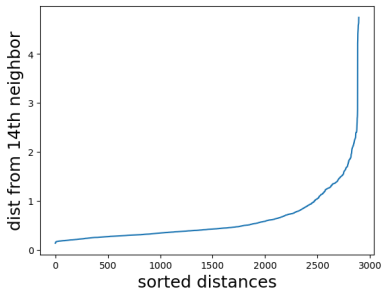


Figure 30: K-Distances DBScan

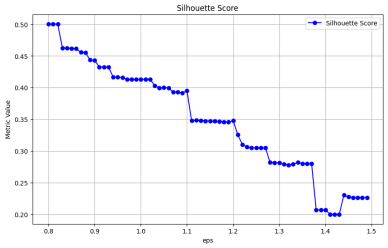


Figure 31: Silhouette DBScan

Figure 32: Proximity Matrix DBScan

For the DBscan algorithm, we use the K-Distance plot (figure 30) to tune epsilon and adjust the value based on the silhouette score (figure 32). In the proximity matrix, the top-left portion shows the outliers detected, which can be ignored. The rest of the matrix is quite similar to those from K-Means and X-Means without the vertical and horizontal lines with high scores, suggesting that DBscan is likely capturing similar behavior to these methods while it is doing a better job in excluding the outliers from these clusters.

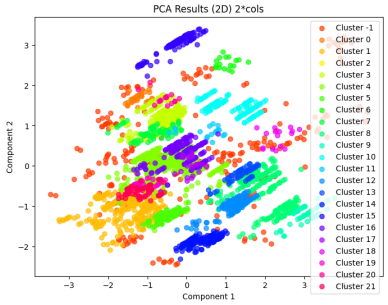


Figure 33: PCA DBScan

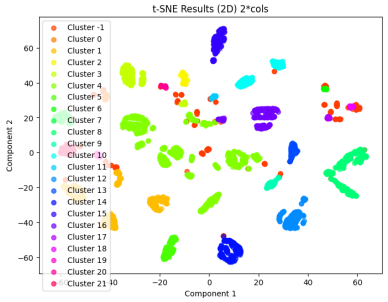


Figure 34: T-SNE DBScan

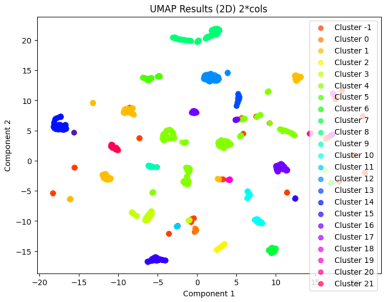


Figure 35: UMAP DBScan

This conclusion is further supported by the visualization plots with PCA, T-SNE and UMAP (figures 33, 34, 35), which closely resemble the results from K-Means and X-Means. Therefore, it is likely that these algorithms are capturing the same or similar well-defined patterns. Also DBscan identifies few outliers that are actually far from other points (figures 33, 35) that belongs to some clusters, suggesting that we tuned enough well the parameters.

3.3 Interpretation of Results

For the interpretation of the results we focus our analysis only on the algorithms that shows better results. In particular we focus only on K-means, X-means and DBscan since the proximity matrix, silhouette, and visualizations are quite good. Further more we avoid interpreting all the results and we focus our analysis only on the most populated clusters and some special clusters that provides meaningful and interesting properties.
To interpret the results, we take into consideration the number of points, the centroids (for K-Means and X-Means) or mean (for DBSCAN), and the standard deviation over the centroids or mean (for DBSCAN).

3.3.1 Standard Deviation Analysis

Feature	Std Dev
Avg Age by URL	0.90
Min Age by URL	1.27
Year	13.33
Points	83.84
Startlist Quality	370.21
Avg Delta by URL	395.59
Length	51691.65

Figure 36: Standard deviation K-Means

Feature	Std Dev
Avg Age by URL	1.09
Min Age by URL	1.26
Year	13.22
Points	79.45
Startlist Quality	382.49
Avg Delta by URL	429.32
Length	56278.27

Figure 37: Standard deviation X-Means

Feature	Std Dev
Avg Age by URL	1.02
Min Age by URL	1.20
Year	11.52
Points	66.99
Startlist Quality	399.67
Avg Delta by URL	487.70
Length	40313.10

Figure 38: Standard deviation DBscan

If we look to the standard deviations tables of these algorithms (figures 36, 37, 38), we can see that values are very similar among the different algorithms. In all cases we have:

- Low variation on Avg_age and Min_Age so we can say that these features doesn't play significant role in cluster differentiation and so they are not useful in cluster interpretation.
- Moderate variation on Year and Points so we can say that these two features play a small role in cluster differentiation and so they can be used to reinforce the pattern found or to find secondary patterns.
- High variation on Avg_Delta, Startlist_Quality and Length so we can say that clusters differentiate mainly on these features and so they are the most source of information to understand which behavior are capturing the clusters.

3.3.2 K-means

Cluster	Count
0	670
1	369
2	250
3	144
4	152
5	153
6	234
7	147
8	119
9	127
10	384
11	146

Figure 39: Cluster points

Cluster	Year	Length	Startlist Quality	Avg Age by URL	Avg Delta by URL	Min Age by URL	Points
0	2012.14	176896.57	1009.31	27.56	373.32	22.03	75.39
1	1992.73	179696.21	1589.95	28.30	394.68	20.00	99.59
2	2005.04	182461.20	870.41	28.61	1131.02	21.69	73.26
3	1996.69	177395.83	1665.80	27.83	862.66	23.00	98.96
4	2019.08	168831.58	1684.99	29.71	19.90	21.00	100.00
5	2012.11	239203.99	985.75	28.73	671.89	21.33	266.67
6	2008.88	27358.97	1007.32	28.35	430.25	21.59	70.60
7	2004.58	173573.47	1729.46	25.85	297.07	20.67	98.91
8	2019.05	159314.87	764.69	28.59	891.72	19.81	48.49
9	1987.61	173268.50	844.17	28.14	1435.27	23.00	80.00
10	2017.63	176218.23	890.41	29.06	158.45	21.02	58.71
11	1994.76	191600.68	1014.73	29.01	322.60	22.00	80.00

Figure 40: Mean (Centroids)

- Cluster 0: Likely represents mid-tier or regional races, with moderately skilled cyclists, younger participants, and mid-range race lengths.
- Cluster 1: Likely represents prestigious, long-standing races with elite athletes and slightly older participants. These races are likely to be historical events with longer race lengths and high point values.
- Cluster 2: Likely represents mid-level stage races or classics with challenging terrains that generate large time gaps with professional but not elite cyclists.
- Cluster 5: Likely represents top-tier, competitive races, likely with older, more seasoned cyclists. These races have a high startlist quality and attract elite cyclists.
- Cluster 10: Likely represents moderately prestigious, longer endurance races that attract fairly experienced cyclists. The events are competitive but not elite, with moderate variability in race outcomes.

3.3.3 X-Means

Cluster	Count
0	490
1	194
2	191
3	119
4	250
5	127
6	48
7	93
8	12
9	98
10	96
11	24
12	128
13	214
14	146
15	144
16	73
17	57
18	76
19	315

Figure 41: Cluster points

Cluster	Year	Length	Startlist Quality	Avg Age by URL	Avg Delta by URL	Min Age by URL	Points
0	2014.79	178389.59	990.31	27.70	478.94	21.97	73.94
1	2005.41	173927.32	1069.70	27.06	79.84	22.12	80.10
2	2017.79	181252.36	850.97	28.47	232.13	20.48	65.55
3	2019.05	159314.87	764.69	28.59	891.72	19.81	48.49
4	2005.04	182461.20	870.41	28.61	1131.02	21.69	73.26
5	1987.61	173268.50	844.17	28.14	1435.27	23.00	80.00
6	2018.13	230945.83	893.40	28.83	103.74	21.77	236.46
7	2013.66	239735.48	990.59	28.70	1045.12	20.63	271.51
8	1976.08	268117.50	1317.58	28.59	51.94	25.00	350.00
9	2017.08	175420.41	1079.73	29.63	136.43	21.12	50.00
10	2017.94	166966.67	776.41	29.65	33.22	22.03	54.22
11	2019.13	82441.67	1684.33	29.71	19.90	21.00	100.00
12	2019.07	185029.69	1685.11	29.71	19.90	21.00	100.00
13	2009.75	27485.75	960.74	28.35	425.78	21.74	68.50
14	1994.76	191600.68	1014.73	29.01	322.60	22.00	80.00
15	1996.69	177395.83	1665.80	27.83	862.66	23.00	98.96
16	2012.95	170584.93	1688.55	25.96	596.91	21.00	100.00
17	1991.95	178957.89	1934.84	25.77	0.00	20.02	100.00
18	1989.54	61026.97	1537.80	28.27	378.58	20.03	98.16
19	1993.99	197613.97	1597.17	28.29	401.36	20.00	99.52

Figure 42: Mean (Centroid)

X-Means provides a more detailed and nuanced view of the data. It captures more diversity in race types than K-Means, with clusters ranging from shorter races with young, less experienced cyclists to longer, high-reward races with elite athletes. We can interpret individual clusters and identify some interesting behaviors:

- Cluster 0: Likely represents mainstream, mid-level professional races that are well-attended by competitive cyclists.
- Cluster 1: Likely represents mid-tier professional races, possibly national tours or regional classics that attract competitive fields.
- Cluster 2: Likely represents emerging or lower-tier races often used by younger riders to build experience.
- Cluster 4: Features medium-level races with significant gaps, likely in hilly or mountainous terrain.
- Cluster 8: Likely represents historic, long-distance, elite-level races with tight competition and maximum prestige.
- Cluster 19: Likely represents highly competitive, long-established races, potentially Grand Tour stages or famous classics.

3.3.4 DBSCAN

Cluster	Count
0	30
1	331
2	55
3	155
4	635
5	99
6	17
7	73
8	234
9	52
10	124
11	18
12	152
13	71
14	144
15	127
16	168
17	21
18	16
19	28
20	15
21	69
-1	261

Figure 43: Cluster points

Cluster	Year	Length	Startlist Quality	Avg Age by URL	Avg Delta by URL	Min Age by URL	Points
0	2012.87	166683.33	985.80	28.71	1505.49	22.00	50.83
1	2017.61	148035.50	897.18	29.32	141.69	21.37	50.00
2	2000.89	186436.36	899.76	29.74	1140.96	21.00	78.36
3	2006.42	168517.10	881.03	28.15	991.83	22.00	78.94
4	2011.00	162118.90	1008.32	27.59	349.70	21.88	78.38
5	2017.84	187454.55	862.67	28.28	204.05	20.00	80.00
6	2017.59	258847.06	818.12	29.05	1409.83	22.00	275.00
7	2016.49	194500.00	1099.47	27.66	624.35	23.00	50.00
8	1984.13	172704.06	1572.43	28.32	327.37	20.00	100.00
9	1991.71	191623.08	1942.50	25.76	0.00	20.00	100.00
10	1997.39	195971.77	1660.47	27.83	869.46	23.00	99.27
11	1996.56	37633.33	1778.06	27.83	869.46	23.00	96.67
12	2006.97	171512.50	1600.82	28.26	511.65	20.00	98.09
13	2012.93	175105.63	1686.61	25.96	596.91	21.00	100.00
14	2019.05	176776.39	1685.79	29.71	19.90	21.00	100.00
15	1987.61	173268.50	844.17	28.14	1435.27	23.00	80.00
16	1994.71	170389.29	1008.48	29.02	324.43	22.00	80.00
17	2018.48	199380.95	747.19	29.62	102.09	22.00	225.00
18	2008.88	246500.00	1027.38	28.17	791.27	20.00	275.00
19	2020.07	162495.71	818.61	28.40	1147.93	19.00	50.00
20	1993.60	175853.33	742.47	28.45	1358.28	21.00	50.00
21	2017.84	170973.91	822.74	28.75	711.76	20.00	50.43

Figure 44: Mean

DBSCAN captures the temporal evolution of cycling races, separating modern elite events from historical classics while also identifying regional and developmental races with distinct competitive dynamics. In particular, we can see:

- Cluster 1: Likely represents modern shorter races that likely cater to regional or national riders. The low point values suggest less prestigious races that may focus on talent development or training events.
- Cluster 8: Likely represents prestigious historical races. The high startlist quality and points reflect elite-level competition, while the moderate gaps suggest well-organized strategic racing dynamics.
- Cluster 14: Likely represents modern, elite-level races with extremely competitive fields.
- Cluster 15: Likely represents older stage races with selective terrains and prestige that attracted strong fields but not elite-level cyclists.

3.4 Comparisons of Results

- K-Means: Captures broad patterns in race types, focusing on grouping races based on race length, cyclist experience, and overall race prestige. It tends to form clusters with more generalized characteristics, often grouping mid-tier and elite races into distinct categories.
- X-Means: Provides a more detailed, nuanced clustering than K-Means. It captures a wider diversity of race types, including emerging and lower-tier races. X-Means identifies a more refined differentiation between various race levels, such as historical prestige and elite competition.
- DBSCAN: Focuses on capturing temporal and competitive dynamics of races, distinguishing between modern elite events and historical classics. It also identifies regional or developmental races with distinct characteristics. DBSCAN's clustering is less sensitive to predefined numbers of clusters, allowing it to detect outliers and varying density in race types.

4 Classification

To solve the classification task assigned to us, we started by deciding the metrics that were interesting for a possible use case, and then tried different Machine Learning (ML) techniques in order to classify wheter a cyclist would reach a top-20 position in a race.

To address the task, we experimented with several models: a Support Vector Machine (SVM) with a linear kernel as an initial baseline, a neural network (NN) consisting of 32 neurons with sigmoidal activation functions, a Random Forest model (RF), and an XGBoost model (XGB).

All models were trained using a set of features comprising both categorical and numerical variables. The categorical features include *profile*, *season*, and *continent*, while the numerical features consist of *length*, *startlist quality*, *BMI*, *age*, and *points*.

4.1 Experimental setting

Hyperparams	Range	Final val.
C	$[10^{-5}, 10]$	10^{-1}

SVM linear kernel hyperparameters ranges.

Hyperparams	Range	Final val.
# estimators	$[200, 400]$	300
max features	$\{0.5, 0.75, sqrt\}$	0.75
max depth	$[10, 50]$	25

RF hyperparameters ranges.

Hyperparams	Range	Final val.
subsample	$[0.8, 1]$	1
# estimators	$[500, 8000]$	2500
min child weight	$\{1, 3\}$	1
max depth	$[6, 15]$	10
learning rate	$[0.3, 1]$	0.3
γ (regulariz. par.)	$\{0, 1\}$	0

XGB hyperparameters ranges.

Hyperparams.	Range	Final val.
learning rate	$[10^{-3}, 10^{-1}]$	10^{-3}
weight decay	$\{10^{-3}, 10^{-2}\}$	10^{-3}
batch size	$\{32, 128, 256\}$	128
epochs	$\{5, 7, 10\}$	7

NN hyperparameters ranges.

Table 1: Hyperparameters ranges for the models studied in the report. When using square brackets we refer to an interval, while when using curly brackets we show all the parameter values tried.

In defining the appropriate classification metric for this task, we considered potential practical applications of the model. For instance, the model could inform decisions such as selecting new cyclists to recruit for a team, allocating team resources to prioritize cyclists with higher winning probabilities, or strategizing for upcoming races. These scenarios place particular emphasis on the **accurate identification of top-20 cyclists** and assign **significant costs to false positives**. Thus, misclassifying a cyclist as a top-20 performer when they are not could lead to substantial financial and strategic impact.

To address this task, we selected the Fβ-score ($\beta = 0.5$) on the top-20 class as the optimization metric. The choice of $\beta = 0.5$ reflects our intention to prioritize precision over recall, giving precision double more importance than recall, in order to minimize the impact of false positives. Since our primary focus is on the performance of the top-20 class, unless explicitly stated otherwise, we report metrics pertaining exclusively to this class throughout this section.

To evaluate the models and choose the best one, after separating the dataset in training and test sets as required, we employed 5-fold cross validation on our training set. The hyperparameters ranges of each model studied are reported in Table 1, alongside the final hyperparameters of the selected classifiers. The final values of the hyperparameters reported in the tables remain fixed for the totality of this section, as fixing them makes for easier comparisons.

As our approach involved randomized searches, it is challenging to detail the specific parameter values and combinations evaluated. However, our general methodology consisted of initially assessing the model's performance using default parameter settings and subsequently adjusting the values based on those that yielded the best results during the search process.

To address the issue of class imbalance within the dataset, as the top-20 class makes for 16% of the training set and 14% of the test set, we experimented with both under-sampling and over-sampling techniques. Additionally, to manage missing values, especially in the case of SVM and NN as they cannot handle them, we investigated several simple imputation strategies and assessed their influence on the predictive accuracy of the models.

For the functionality of each parameters, we refer back to the Scikit-Learn documentation for LinearSVM and RandomForestClassifier, to the Keras documentation for the NN and to the XGBoost library. The notebooks also feature a technically functional Gaussian kernel SVM training pipeline. However, we observed that this approach demanded an excessively time to train: after several hours of processing, the training on a single fold was still incomplete.

Model	Strategy	Precision	Recall	F1-score	F0.5-score	Accuracy
SVM	Over.	0.2377	**0.4804**	0.3180	0.2644 ± 0.0013	0.6659
SVM	Under.	0.2380	0.4801	0.3182	0.2647 ± 0.0011	0.6664
SVM	Balanced	**0.2381**	0.4803	**0.3184**	**0.2648 ± 0.0011**	**0.6665**
RF	Over.	0.448	0.470	**0.459**	0.452 ± 0.002	0.820
RF	Under.	0.319	**0.651**	0.428	0.356 ± 0.001	0.718
RF	Balanced	**0.527**	0.371	0.435	**0.486 ± 0.002**	**0.844**
NN	Over.	**0.253**	**0.573**	**0.351**	**0.285 ± 0.003**	**0.656**
NN	Under.	0.242	0.591	0.343	0.274 ± 0.002	0.633
XGB	Over.	**0.429**	0.450	**0.439**	**0.433 ± 0.002**	**0.814**
XGB	Under.	0.296	**0.677**	0.412	0.334 ± 0.002	0.687
XGB	Balanced	0.425	0.455	**0.439**	0.430 ± 0.003	0.812

Table 2: Results of the experiments to handle the class imbalance in the dataset. All scores, accuracy excluded, are reported w.r.t. the top-20 class on validaiton data. Results in **bold** are the best for the specific model, while those in blue are the best overall.

4.2 Sampling impact

We initially observed during the training of the ML models that not addressing the class imbalance resulted in extremely poor performance. Although the results of these cases are excluded, we observed metrics approaching zero across all evaluation measures related to the top-20 cyclists. Consequently, we investigated various techniques to address the class imbalance present in the dataset.

To address this issue, we investigated three primary approaches: random oversampling, random downsampling, and leveraging model-specific parameters to handle class imbalance. The model-specific parameters included *class_weight* for SVM and RF, set to "balanced" to appropriately weight the two classes relative to each other, and *scale_pos_weight* for XGB, fixed to the ratio (#non-top-20)/(#top-20), as recommended by the XGBoost documentation. While Keras neural networks theoretically offer a hyperparameter for handling class imbalance, our experiments with various values failed to yield minimally satisfactory results.

The notebooks also include a SMOTENC pipeline for oversampling. However, we were unable to obtain results using this technique due to its substantial memory requirements, which exceeded our available resources (more than 8GB of RAM were required).

All results were obtained using median imputation for the *cyclists_age*, *height*, and *weight* columns. The *height* and *weight* columns were used solely to compute *BMI* and subsequently dropped. Missing values in the profile column were left unaltered for RF and XGB or imputed as 0 for SVM and NN. Numerical columns were standardized using z-standardization for SVM and NN, while categorical columns were one-hot encoded for SVM, NN, and RF.

Table 2 presents the results of our analysis. As observed, RF achieves the highest F0.5-score among all models, along with the highest precision. Notably, all models exhibit performance degradation when undersampling is applied compared to other techniques that either maintain or increase the amount of data available for training. Random oversampling leads to better performance for XGB and NN, while RF and SVM benefit more from balancing class weights. This effect is particularly pronounced in RF, where a significant drop in F0.5-score is observed with oversampling and undersampling, in contrast to SVM, which demonstrate relatively consistent performance across techniques.

4.3 Imputation impact

Another critical aspect to address in our dataset was the presence of missing values. This was particularly important since models like SVM and NN cannot handle missing values directly,unlike RF and XGB.

The experiments for this analysis were conducted using random oversampling for NN and XGB, while class weights were applied for SVM and RF. These approaches were selected based

Model	Strategy	Precision	Recall	F1-score	F0.5-score	Accuracy
SVM	Median	**0.238**	0.480	0.318	**0.265 ± 0.0011**	**0.667**
SVM	Median + Most freq.	0.208	**0.566**	0.305	0.239 ± 0.001	0.581
SVM	Mean	0.237	0.482	**0.318**	0.264 ± 0.001	0.664
RF	No imp.	0.518	**0.374**	0.434	0.480 ± 0.002	0.842
RF	Median	0.527	0.371	0.435	0.486 ± 0.002	**0.844**
RF	Median + Most freq	0.524	0.364	0.429	0.481 ± 0.002	0.843
RF	Mean	**0.529**	0.373	**0.437**	**0.488 ± 0.005**	**0.844**
XGB	No imp.	**0.433**	**0.453**	**0.443**	**0.437 ± 0.001**	**0.815**
XGB	Median	0.429	0.450	0.439	0.433 ± 0.002	0.814
XGB	Median + Most freq	0.427	0.450	0.438	0.432 ± 0.002	0.813
XGB	Mean	0.429	0.450	0.439	0.433 ± 0.003	0.814
NN	Median	**0.253**	0.573	**0.351**	**0.285 ± 0.003**	**0.656**
NN	Median + Most freq.	0.245	**0.575**	0.344	0.277 ± 0.005	0.644
NN	Mean	0.248	0.591	0.349	0.280 ± 0.003	0.643

Table 3: Results of the experiments to handle the missing values in the dataset. All scores, accuracy excluded, are reported w.r.t. the top-20 class on validaiton data. Results in **bold** are the best for the specific model, while those in blue are the best overall.

Model	Precision	Recall	F1-score	F0.5-score	F0.5-score val.	Accuracy
Always top-20	**0.14**	**1.00**	**0.24**	**0.17**	**0.19 ± 3.717**	0.14
Always non top-20	0	0	0	0	0 ± 0	**0.86**
Rand. uniform	**0.14**	0.51	0.22	**0.17**	0.18 ± 0.001	0.50
Rand stratified	**0.14**	0.16	0.15	0.14	0.16 ± 0.002	0.75
SVM	0.16	0.19	0.17	0.16	0.27 ± 0.0011	0.74
RF	**0.31**	0.09	0.14	0.21	**0.49 ± 0.006**	**0.85**
XGB	0.17	0.29	0.22	0.19	0.44 ± 0.001	0.70
NN	0.22	**0.42**	**0.29**	**0.24**	0.29 ± 0.003	0.71

Table 4: Test set performances for the models investigated trough our report. Validation F0.5-score is also included for reference.

on their effectiveness in the previous analysis of data imbalance techniques. Scaling and one-hot encoding were applied in the same manner as in the prior experiments.

Table 3 presents the results of the analysis. For numerical columns, both mean and median imputation were evaluated, while for the *profile* column, imputation using the most frequent category was explored.

Our first observation is that imputing the profile column negatively impacts performance, as it yields the lowest F0.5-score across all models. This outcome is primarily attributed to the fact that using the most frequent values fails to accurately reflect the true distribution of the profile data, undermining the model performance.

For SVM and NN, median imputation emerges as the most effective approach, although its performance is very similar to mean imputation in terms of F0.5-score, particularly when accounting for the variance in the measurements.

RF notably benefits from imputing numerical values, with both median and mean imputation yielding better overall performance compared to not performing imputation. In contrast, XGB performance is slightly hindered by any kind of imputation.

During our experiments, we also explored using k-NN imputation to address missing values. However, similar to SMOTENC, the high memory consumption of this approach prevented us from effectively evaluating its validity.

4.4 Test set results

Throughout this section, we have had the opportunity to explore the performance of the different models, providing insight into their potential test set performance. In this final part of the section, we aim to evaluate the best model configuration identified for each of the four ML

models on the test set. For this subsection, each model pipeline corresponds to the configuration that achieved the best results on validation (the ones highlighted in bold in Table 3).

All ML models were refitted on the full training set prior to evaluation on the test set. Specifically for the NN, to avoid problems regarding the choice of the number of epochs, due to the different amount of data used during 5-fold validation and the final refitting, we separated 20% of the training set as a validation set. This allowed us to employ early stopping with a patience of 3 epochs, which resulted in training for 7 epochs.

Table 4 presents the results of the four best models on the test set and compares them with four naive baselines. *Always top-20* refers to always classifying as top-20, *Always non top-20* is the opposite, *Rand. uniform* involves randomly guessing with a 50% probability for each class, and *Rand Stratified* involves guessing based on the probability distribution of the classes in the training set.

A first important take can be made: overall there is a significant drop between validation and test scores for each of the models. While the NN still manages to achieve decent results when compared to the validation score, achieving the best test set results in terms of F0.5-score, the other models experiment a huge drop in performance, especially RF which gets overtaken by NN.

This phenomenon suggests the presence of covariate shift between the training and test sets. This is a plausible explanation, given that the test set consists solely of races from 2022 onwards. Considering the changing trends in average weight and height of cyclists, as shown in Figure 3, it is not unreasonable to expect a distributional shift between the training and test sets. However, the NN's performance undermines this hypothesis. Even so, it may have been beneficial to better account for the fact that the goal is to classify future races. For instance, instead of using 5-fold validation, which randomly splits the training set without considering the year of the competition, we could have used a validation set that included races from the most recent years within the training set, similar to the original split. It is also important to note that if there is indeed a trend in the dataset, reframing the task and employing other models designed to capture such trends could potentially yield more beneficial results.

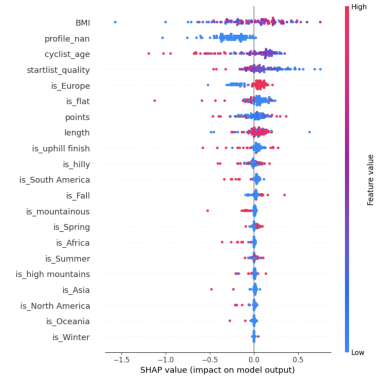
Indeed, an important observation that we have only now address is the consistently poor performance of all the models. All models barely outperform the random baselines, with SVM even performing below some of them, denoting the limited predictive power of the models. This suggests that the task would likely have benefited from additional features, maybe representing the past performances of cyclists in other, preferably similar, races. Alternatively, reframing the task as a time series forecasting problem—where the goal is to predict future performance based on a cyclist's past placements—could potentially yield more favorable results.

5 Explainability

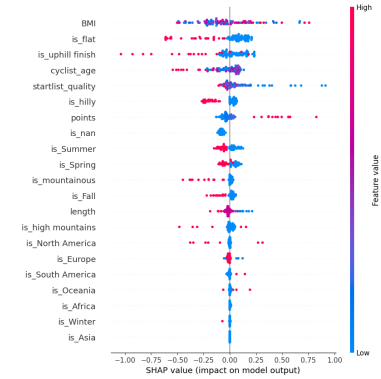
The final section of our report focuses on explaining the models used in the classification task, specifically the RF and NN. We limited the analysis to these models as RF demonstrated the best validation performance, while the NN is the best on the test set. Comparing the RF explanations with those of the NN provides insights into the biases of both models, helping us understand the significant performance drop observed in the RF relative to the NN on the test set.

Since the explainability methods require a small dataset and focus on local explanations, we limited the analysis to 100 samples from the training and test sets.

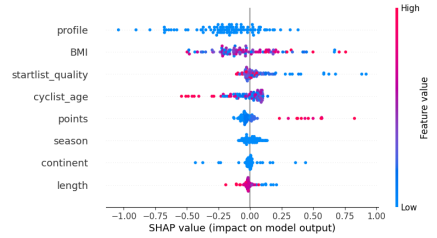
To construct the datasets, we used K-means clustering (with categorical columns one-hot encoded) to identify 1,000 centroids. For each centroid, the closest sample in the original dataset was selected using Euclidean distance. From these 1,000 samples, we randomly selected 100 to form the final datasets for this analysis. Opting for 1,000 initial centroids ensured the inclusion of top-20 positions, in contrast to using directly 100 centroids. The reduced training



(a) Beeswarm plot of SHAP values for RF model on sampled test set.



(b) Beeswarm plot of SHAP values for NN model on sampled test set.



(c) Beeswarm plot of SHAP values, with aggregated impact of categorical features, for NN model on sampled test set.

dataset is exclusively used for SHAP, while other methods use the full training set. However, all explanations presented are based on samples from the reduced test set.

5.1 General behavior of the models

For the SHAP explanations, the sampled training set was used as the background distribution, and explanations were computed on the sampled test sets. The primary results are presented in Figures 45a and 45b, where features are ordered by their mean contribution, calculated as the mean absolute value of their contributions across the sampled test set.

For both models, BMI, cyclist age, and startlist quality emerge as highly influential features, with race profile exerting the greatest impact in both cases.

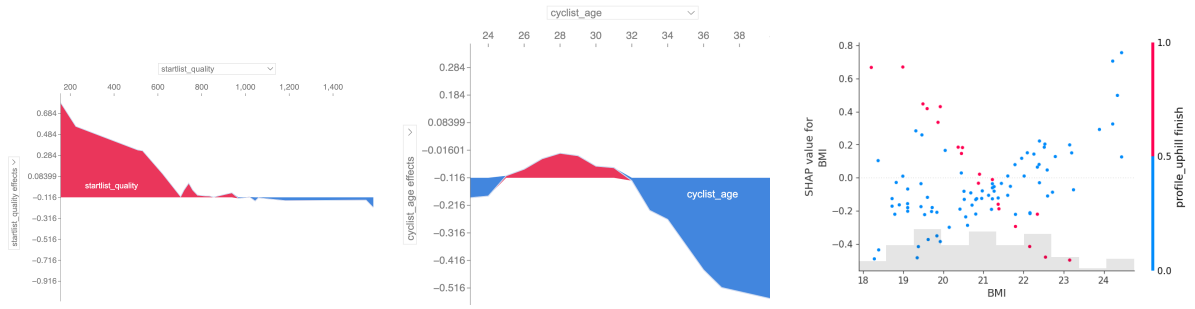
Regarding race profiles, the RF model heavily penalizes races where the profile is not missing, while the NN assigns relatively little negative weight to missing profile values. This discrepancy appears to be a key reason for the RF model's poorer performance on the test set, where profile data is consistently available. This outcome suggests that removing rows with missing profile values from the training and validation sets could have improved performance.

However, given the occurrence of missing values in this column, retaining them during training was a reasonable choice to ensure the model could handle such cases, although it has negatively impacted performance.

The features representing the race season exhibit unusual behavior. Although their overall impact is not negligible for either model, the RF model tends to assign a positive weight to samples from a specific season, while the NN shows the opposite trend. This creates an inconsistent pattern, which we speculate, as seen in Figure 45c, results from the features being unimportant for classification. We propose that the negative contribution from the active season feature is counterbalanced by the positive contributions from the inactive season features, leading to an overall low impact.

Regarding cyclists' continents, the NN model typically assigns non-zero scores for cyclists from non-European regions, while it shows the opposite behavior for European cyclists. In contrast, the RF model penalizes non-European continents and assigns positive SHAP values for European cyclists. This suggests that excluding the continent feature from the models could have mitigated potential bias, as the data is disproportionately represented by European cyclists.

A last interesting observation concerns the points associated with a race. The NN model appears to associate higher points with a greater chance of winning, which is counterintuitive, as typically, one might expect that races offering more points would be more competitive and, therefore, more difficult to win.



(a) Startlist quality impact for different feature values

(b) Cyclist age impact for different feature values

(c) BMI impact w.r.t. whether a race has an uphill finish

Figure 46: Plots observing the behaviors of different features for the NN model. RF plots are omitted as they show similar behavior.

	cyclist_age	profile	points	length	startlist_quality	BMI	season	continent	classified
0	27.0	hilly	50	169400.0	981	20.971172	Summer	Europe	fp
1	31.0	uphill finish	100	144900.0	1584	19.379197	Summer	Europe	fn

Figure 47: Samples considered in the analysis. The first sample is a false positive for the RF, while the second is a false negative. Both are correctly classified by the NN.

5.1.1 Feature behavior analysis

While Figures 45a and 45b provide an overview of the model's general behavior, they do not clearly show how specific outcomes are impacted by high or low feature values, as the SHAP values do not exhibit a distinct separation. To gain more insight, we further investigated the SHAP values for cyclist age, BMI, and startlist quality, as these features exhibited the most ambiguous behavior. Figure 46 offers a clearer understanding of how the model interprets these features.

Regarding startlist quality, Figure 46a illustrates that as the quality of the starting list increases, the Shapley values decrease. This captures the relationship that higher-quality participants in a race make it less likely for a given cyclist to win.

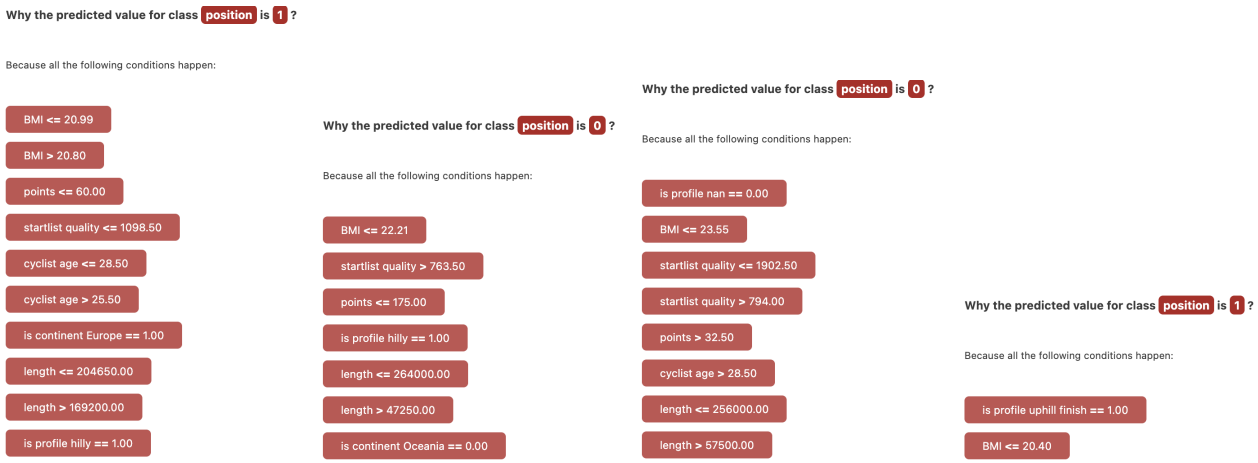
For cyclist age, Figure 46b shows that the model assigns positive value to ages between 25 and 32 years, indicating that cyclists within this age range are more likely to win the race.

Finally, Figure 46c reveals how BMI influences model predictions, showing two distinct trends in the SHAP values based on the race profile. For races with a flatter profile, the model assigns higher values to BMI, favoring cyclists with greater muscle mass. Conversely, for races with an uphill finish, the model favors lower BMI values, as lighter cyclists have an advantage in mountainous terrain due to their reduced weight. This aligns with real-world observations, where lighter cyclists perform better in steep races, while those with higher BMI excel in flatter races by leveraging their increased muscle mass.

5.2 Local factual and counterfactual explanations

In the final part of the explainability section, we analyze the behavior of the models on specific samples, focusing on providing both rule-based and counterfactual explanations. The analysis is limited to two samples, shown in Figure 47. The first sample is a false positive for the RF model, while the second is a false negative. Both samples are correctly classified by the NN. The sample selection was driven by the goal of understanding the factors, beyond those identified in the SHAP analysis, in order to better explain why the NN outperforms the RF on the test set.

To select the samples, we noted that the RF model only produced two false positives on



(a) (b) (c) (d)

Figure 48: a Explanation for false positive explanation of RF b Explanation of the NN for the false positive of RF c Explanation for false negative of RF d Explanation of the NN for the false negative of RF

the restricted test set, one of which was also a false positive for the NN and therefore not considered. The false negative sample instead was just the first that met our selection criteria.

To obtain rule-based explanations, we employed LORE with default parameters, using the entire training set as support data. The factual explanations for both models are shown in Figure 48.

For the false positive, the RF model provides a generally sensible explanation (Fig. 48a), but its reasoning is driven by an unusually restricted BMI range, which seems odd, and the fact that the cyclist is European. The NN model (Fig. 48b) in this regard uses an upper-bound on the BMI as the rationale for why the cyclist did not achieve a top-20 position, suggesting the BMI was too low for that race profile and startlist quality.

For the false negative, the NN explanation (Fig. 48d) is quite simple, relying on BMI and the mountainous race profile with an uphill finish, further confirming our previous assumptions. In contrast, the RF model overlooks the race profile and attributes the cyclist's poor performance solely to a low BMI, which, as previously explained, is actually an advantage in such races.

5.2.1 Counterfactual explanations with LORE

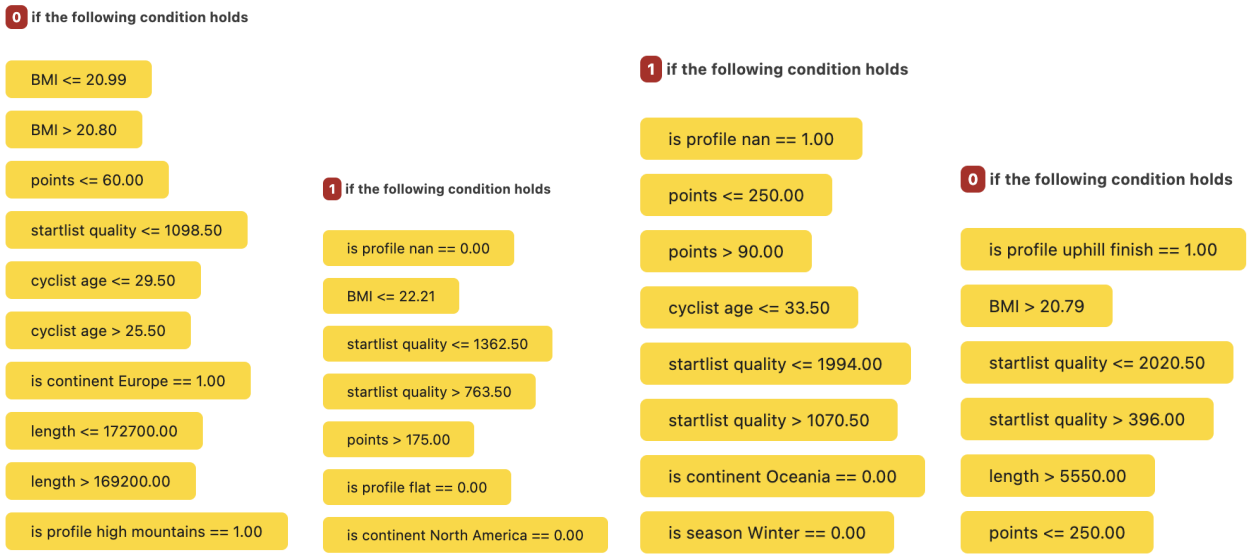
Continuing our analysis, we examined counterfactual explanations for the two samples considered, using both LORE for rule-based counterfactuals and DiCE.

Examples of LORE counterfactuals are shown in Figure 49. For the false positive, the RF model suggests that changing the race profile from hilly to high mountains would prevent the cyclist from reaching the top-20 position (Fig. 49a). The NN, instead, proposes that if the race points were higher, the cyclist would have a better chance of winning, which reflects the nonsensical pattern previously highlighted (Fig. 49b).

For the false negative, the RF model (Fig. 49c) suggests that the cyclist could have reached a top position if they were from Oceania and the race were in Winter, which is highly illogical. In contrast, the NN explanation (Fig. 49d) indicates that the cyclist would have needed a higher BMI to avoid losing.

5.2.2 Counterfactual explanations with DiCE

To obtain the DiCE counterfactuals, we explored all three model-agnostic generation methods provided by the DiCE library: random generation, genetic algorithms, and KD-tree selection from training data.



(a) (b) (c) (d)

Figure 49: (a) Counterfactual explanation for false positive explanation of RF (b) Counterfactual explanation of the NN for the false positive of RF (c) Counterfactual explanation for false negative of RF (d) Counterfactual explanation of the NN for the false negative of RF

For this analysis, given the significant influence of BMI and age on the final outcome, we fixed these features (along with the cyclist's continent) to address the question: "What changes would a race need to undergo for the cyclist to win or lose?"

Figure 50 presents the counterfactuals based only on random generation, as the KD-tree method surprisingly could not find counterfactuals for both examples without adjusting BMI, even slightly, and the genetic algorithm produced low diversity in the explanations.

For the false positive, the NN suggests that to reach the top-20, we would either need to increase the race points (which, as previously noted, was an important factor for the NN) or significantly lower the quality of the starting list.

The RF, on the other hand, proposes more interesting changes. Since the RF predicts that the cyclist will reach the top-20, the counterfactuals aim to show scenarios where the cyclist would lose. Some examples include increasing the startlist quality and race points, having a steeper race profile (as the model assumes that BMI is unsuitable for such profiles), and, in one case, moving the race from Summer to Spring, which seems to reduce the chances of winning, possibly because the cyclist performs better in hotter conditions.

For the false negative, the NN suggests changing the race profile to a less steep one (and, in one case, lowering the race points) denoting the relatively low BMI. In contrast, the RF proposes increasing the points and making the race profile less steep, which implies that the two models interact with the profile and BMI in different ways: for similar changes, the NN results in the opposite outcome for the cyclist.

Query instance (original outcome : 0)

	cyclist_age	profile	points	length	startlist_quality	BMI	season	continent	position
0	27.0	2.0	50	169400.0	981	20.971172	2	2	0

Diverse Counterfactual set (new outcome: 1)

	cyclist_age	profile	points	length	startlist_quality	BMI	season	continent	position
0	-	-	161.0	-	-	-	-	-	1.0
1	-	-	-	-	-	-	-	-	1.0
2	-	-	-	-	410.0	-	-	-	1.0
3	-	-	223.0	-	-	-	-	-	1.0
4	-	-	174.0	-	-	-	-	-	1.0

Diverse Counterfactual set (new outcome: 0)

	cyclist_age	points	length	profile	startlist_quality	BMI	season	continent	position
0	-	279.0	-	-	1716.0	-	-	-	0.0
1	-	287.0	-	3.0	-	-	-	-	0.0
2	-	-	-	-	654.0	-	1.0	-	0.0
3	-	-	-	5.0	879.0	-	-	-	0.0
4	-	-	-	-	-	-	-	-	0.0

(a)

Query instance (original outcome : 1)

	cyclist_age	profile	points	length	startlist_quality	BMI	season	continent	position
0	31.0	5.0	100	144900.0	1584	19.379198	2	2	1

Diverse Counterfactual set (new outcome: 0)

	cyclist_age	profile	points	length	startlist_quality	BMI	season	continent	position
0	-	-	-	-	-	-	-	-	0.0
1	-	1.0	47.0	-	-	-	-	-	0.0
2	-	3.0	-	-	-	-	-	-	0.0
3	-	4.0	-	-	-	-	-	-	0.0
4	-	-	-	-	-	-	-	-	0.0

Diverse Counterfactual set (new outcome: 1)

	cyclist_age	points	length	profile	startlist_quality	BMI	season	continent	position
0	-	194.0	-	2.0	-	-	-	-	1.0
1	-	194.0	-	2.0	-	-	0.0	-	1.0
2	-	288.0	-	3.0	2011.0	-	-	-	1.0

(b)

Figure 50: (a) Counterfactuals of DICE for false positive of RF. (b) Counterfactuals of DICE for false negative of RF. (On top) The counterafctuals of the NN. (On bottom) The counterfactuals of the RF