

Damien d'Estienne du Bourguet

Raphaël Dussart

Yousef Abouturkia

Gabriel Barbier

## **Report on the Triathlon Performance Analysis and Team Formation Project**

### Introduction

This report outlines the development and implementation of a Python-based application designed to analyze, visualize, and optimize athletic performance in triathlon competitions. The project was commissioned by SportVision in collaboration with the Sports Office (BDS) and ESME, aiming to enrich in-depth analyses for interactive TV formats. The primary objectives were to establish a comprehensive ranking of triathlon performances and propose strategies for forming balanced teams based on various performance criteria.

### Project Requirements

The project requirements were structured around several key functionalities:

#### 1. Data Loading and Preprocessing:

- Import and clean athlete data from a CSV file.
- Normalize performance metrics to ensure consistent comparisons.

#### 2. Multi-Criteria Evaluation:

- Develop a method to calculate a global score for each athlete based on weighted performance indicators.
- Implement filters to select subsets of athletes by discipline, age group, or gender.

#### 3. Sorting Algorithms:

- Implement and compare insertion sort and selection sort algorithms.
- Measure and display the execution times of each algorithm.

#### 4. Team Formation:

- Propose strategies for forming balanced teams using the K-Means algorithm.
- Optimize team distribution based on gender and performance criteria.

#### 5. User Interface:

- Create a dedicated section to display rankings dynamically.
- Develop an intuitive user interface for real-time analysis and team formation simulation.

### Data Definition

The dataset included identification and performance data for each athlete:

- Identification Information: Full name, age group, gender, country, and event status.
- Performance Data: Times for swimming, cycling, running, transitions, and overall finish time, along with rankings by gender, age group, and overall.

#### Score Calculation:

Two approaches were considered for calculating the global score:

1. Weighted Sum: A raw score calculated as a weighted sum of times for each discipline and transition.
2. Normalization and Aggregation: Normalizing times relative to average or best times, followed by aggregation using z-scores.

#### Implementation

##### Data Processing

- Loading and Cleaning: The application reads data from a CSV file, converts times to seconds, and handles missing values. It filters out athletes with invalid finish times or disqualified statuses.
- Score Calculation: Global scores are calculated using adjustable weights for each performance indicator.

##### Sorting Algorithms

- Insertion Sort and Selection Sort: Both algorithms were implemented to sort athletes based on their global scores. Execution times were measured to compare their efficiency.

##### Team Formation

- K-Means Clustering: The K-Means algorithm was used to form balanced teams based on global scores. The number of teams and gender distribution criteria were adjustable.

##### User Interface

- Tkinter GUI: A graphical user interface was developed using Tkinter, allowing users to load data, clean it, calculate scores, sort athletes, measure sorting times, and form teams. The interface includes filters for gender and age groups and displays rankings and team compositions dynamically.

#### Results

- Sorting Algorithm Comparison: Insertion sort and selection sort were compared, with execution times measured for different data volumes.
- Team Formation: Balanced teams were formed using the K-Means algorithm, with options to adjust the number of teams and gender distribution.

## Conclusion

The project successfully met the requirements by developing a modular and intuitive application for analyzing triathlon performances. The implementation of sorting algorithms and the K-Means clustering for team formation provided valuable insights into athletic performance optimization. The user interface facilitated real-time analysis and simulation, enhancing the interactive TV format for SportVision.

## Difficulties encountered:

We had some trouble with the interface of this project, as we had never used tkinter before. So we spent a lot of time figuring out how to implement what we wanted to do. We had problems with the geography of the interface (placement of the buttons), as well as issues with the clickable buttons (issues with the chronological order of execution), which is why we opted for the use of new windows opening each time we clicked on a button.