April 2024

WORK ON SUPER MESSY DATASETS

This project:

Data Cleaning on over "7,000,000" Rows of Data

Prepared By:

SAEED SHIRANI

saeedshirani.github.io
saeedshirani7878@gmail.com

About the Series and the project:

In this series of projects named “Work on Super Messy Datasets,” each time I encounter a dataset with a heavy amount of data (often from real-world sources), I undertake the task of cleaning the dataset and sometimes performing Exploratory Data Analysis (EDA) on it.

This can be a highly challenging task, serving as a great opportunity to test and leverage our skills.

In this particular project, I'll be tackling a very large dataset containing **7,000,000 records of two centuries of marathon matches** held across the globe. My primary concern is to implement changes without compromising data integrity as much as possible.

TABLE OF CONTENTS

THE OVERVIEW OF DATASET

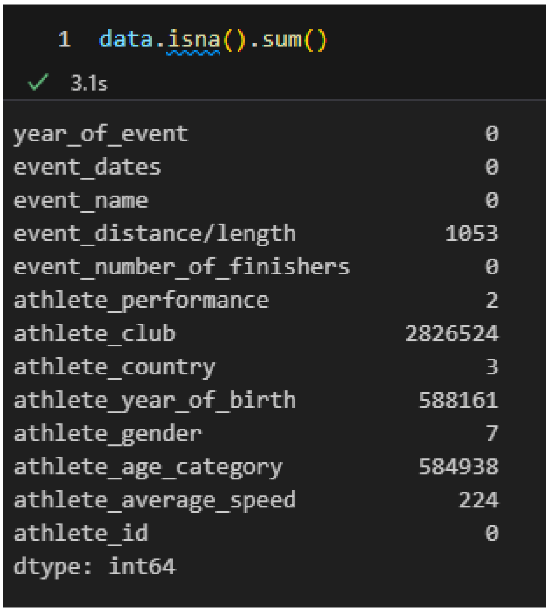
The dataset is available for download at this link:
The big dataset of ultra-marathon running
Additionally, you can find more information about the dataset at this link.

The dataset is created in 13 columns with these names:

1. **Year of event:** No Missing Value
2. **Event Dates:** No Missing Value
3. **Event Name:** No Missing Value
4. **Event Distance/length:** Include missing value
5. **Event Number of finishers:** No Missing Value
6. **Athlete performance:** Include missing value
7. **Athlete club:** Include missing value
8. **Athlete country:** Include missing value
9. **Athlete year of birth:** Include missing value
10. **Athlete gender:** Include missing value
11. **Athlete age category:** Include missing value
12. **Athlete average speed:** Include missing value - unit: km/h or m/h (in most cases)
13. **Athlete ID:** No Missing Value

STEPS OF MY DATA PROCESSING

- **Renaming Columns**
- **Handling Missing Values** (And deciding which rows must get dropped.)

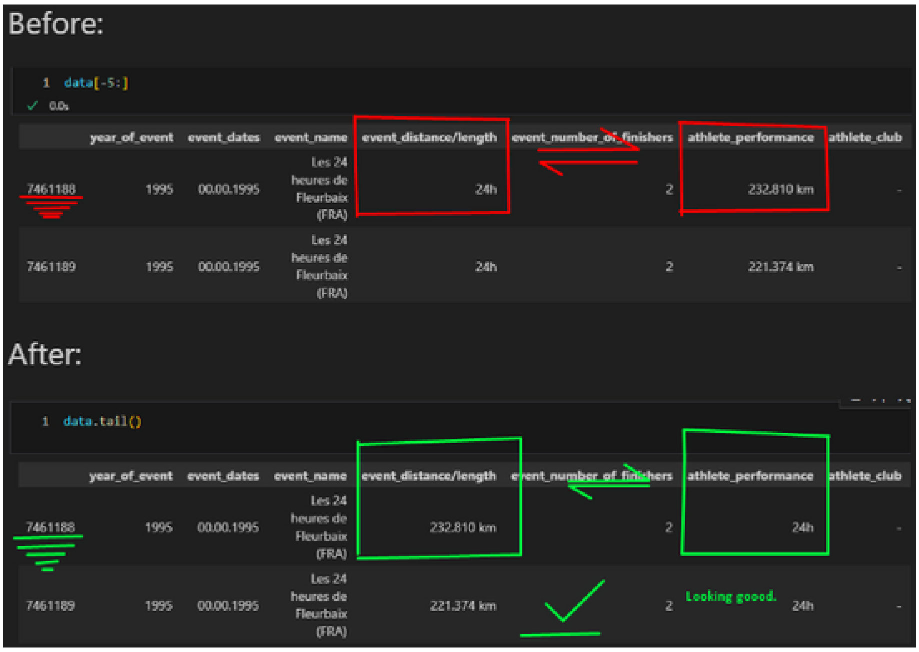


```
data.isna().sum()
```

✓ 3.1s

```
year_of_event                     0
event_dates                       0
event_name                        0
event_distance/length          1053
event_number_of_finishers         0
athlete_performance               2
athlete_club                2826524
athlete_country                   3
athlete_year_of_birth        588161
athlete_gender                    7
athlete_age_category         584938
athlete_average_speed           224
athlete_id                        0
dtype: int64
```

- **DATA SWAP** (BETWEEN "ATHLETE_PERFORMANCE" AND "EVENT_DISTANCE/LENGTH" COLUMNS)



Before:

```
data[-5:]
```

✓ 0.0s

	year_of_event	event_dates	event_name	event_distance/length	event_number_of_finishers	athlete_performance	athlete_club
7461188	1995	00.00.1995	Les 24 heures de Fleurbaix (FRA)	24h	2	232.810 km	-
7461189	1995	00.00.1995	Les 24 heures de Fleurbaix (FRA)	24h	2	221.374 km	-

After:

```
data.tail()
```

	year_of_event	event_dates	event_name	event_distance/length	event_number_of_finishers	athlete_performance	athlete_club
7461188	1995	00.00.1995	Les 24 heures de Fleurbaix (FRA)	232.810 km	2	24h	-
7461189	1995	00.00.1995	Les 24 heures de Fleurbaix (FRA)	221.374 km	2	24h	-

- **CALCULATING AND FILLING YEAR OF BIRTH**

I calculate 'athlete_birth_year' based on year_of_event and athlete age

- **Set suitable Data types**

in this data cleaning process; I just converted mile to kilometer, other changes are very dependent on the nature of our analysis

- **EXPORTING AN ALMOST CLEAN DATASET**

Finally, my ETL code will save a CSV file, named "clean_dataset" in the same directory of the ETL notebook.
You can run a notebook to process the raw dataset and create a final CSV file for your feature analysis.

MAIN CHALLENGES

One of the main problems with this dataset is the inconsistency in data types across columns. There isn't a single data type that can adequately cover all the values in each row of the dataset. This inconsistency arises due to the wide variety of values present in each column.

For instance, in the "Athlete performance" column, we encounter values representing durations in hours, minutes, and seconds. However, some records deviate from this pattern and express durations in terms of days, such as "6 d - 12 h - 4:25:31 h" and so forth.

This lack of uniformity in data representation poses a significant challenge when trying to establish a consistent data type for each column.

Furthermore, in the "Event Dates" column, we observe a diverse range of data formats. Occasionally, it consists of a single date, such as "01.11.2018", while at other times, it includes a date range, such as "03-08.12.2015". This indicates that the match occurred between the dates 03.12.2015 and 08.12.2015.

This inconsistency in date representations further complicates standardizing data types within the dataset.

Another challenge lies within the "Event Distance/Length" column. This column contains the distances that athletes ran in each match. The main issue here is the presence of data in both miles and kilometers, stored in various formats such as "10 km", "10km", "10KM", "10mi", "10Mile", and many other variations.

To address this inconsistency and standardize the data, I will convert all values to the kilometer scale. This will involve parsing the values, identifying whether they are in miles or kilometers, and converting them accordingly to a consistent unit, such as kilometers.

In certain cases, we encounter missing values in the "Year of Birth" or "Athlete Age Category" columns. To mitigate this issue, we can calculate each of these missing variables using another variable if available. However, if both variables are missing for a particular row, we have no choice but to exclude that row from our analysis or processing. This ensures that we maintain data integrity and avoid making inaccurate assumptions or imputations when essential information is lacking.

The column “athlete_average_speed” has **221** missing values which I can not fill them because of wide range of types in column.

RESULTS

The final dataset comprises 1,000,000 rows of data. For feature analysis, we must either select a subset of the data or define our goals and strategies for analysis. Based on the defined conditions, we need to decide the appropriate treatment for each type of value to ensure data accuracy.

For example, in the 'athlete_performance' column, there is a wide range of time accuracy, including seconds, minutes, hours, and days. Our choice of time conversion depends heavily on the required accuracy for our analysis and the goals we have defined.

Also, in our dataset, there were around 150 rows with no reasonable values and inaccurate data. Therefore, there was no choice but to drop these rows.