Project Report GitHub URL

pauldardis/UCD_Data_Analytics_Project_2 (github.com)

*** For security reasons you can only access this link if you are set as a collaborator, please email paul.dardis@bt.com if you require access ***

Abstract

Strava is a popular social fitness platform designed for athletes and fitness enthusiasts. It combines the functionalities of a fitness tracking app and a social network, allowing users to track and analyse their activities while connecting with a community of like-minded individuals.
The basic level of Strava access is free however for more detailed analysis of activity's a substitution is required.
This project focuses on downloading and analysing Strava Activity data by utilizing APIs to extract the data and implementing machine learning algorithms to classify the intensity level of each activity. By leveraging these technologies, the project aims to provide free insights into the user activities and categorize them based on their intensity to assist them with their activity goals.

Data Extraction: Utilise Strava's APIs to download activity data, including details such as duration, distance, elevation, heart rate, and more.

Data Cleansing: Removing abnormal data, reformatting measurement (distance , KPM, etc)

Machine Learning Classification: Applying machine learning algorithms to classify the intensity level of each activity. By training the algorithms so that they can learn to differentiate between low, moderate, and high-intensity activities based on various parameters such as heart rate, pace, and elevation.

Visualization and Insights: data visualization techniques to present the analysed activity data.

Activity Comparison: The software enables users to compare their activities based on intensity levels. This feature allows users to see how their performance varies across different activities, helping them identify patterns and make informed decisions about their training.

Introduction

The Mallorca 312, a cycling race held in April, has been a personal challenge for me since 2017. I have participated every year; however, I've only managed to successfully complete it twice, in 2022 and 2023. Now, I'm undertaking this project to investigate the changes I made in those years that enabled me to complete the race and to identify areas for further improvement in future races.

By utilizing machine learning, I aim to determine the relative effort for each of my training activities. This approach will provide a more objective assessment compared to relying solely on personal perception, which can be subjective and influenced by mood.

My plan is to categorize my training activities into three levels: hard, medium, and low. This categorization will help identify patterns, trends, and areas for improvement in my training routine. Using the data from my past activities, it is hoped that the machine learning algorithms can provide valuable feedback and insights into how to optimize my training for better performance in the Mallorca 312 and other cycling challenges.

Dataset

The dataset contains a record of my activities on Strava dating back to 2011 [Paul Dardis IE | Strava Cyclist Profile](). The reason for choosing this dataset was to eliminate any concerns around data protection and more importantly is develops my skills utilizing the Strava API and analysing data, with the aim of eventually expanding the project to include the data of other members within my club. The dataset encompasses the following information.

Index	Column Name	Description	Non-Null Cou	Dtype
1	resource_state		3648	int64
2	name	The name of the activity.	3648	int64
3	distance	In meters.	3648	object
4	moving_time	The activity's moving time, in seconds	3648	float64
5	elapsed_time	In seconds	3648	int64
6	total_elevation_gain	The activity's total elevation gain.	3648	int64
7	type	Type of activity. For example - Run, Ride etc.	3648	float64
8	sport_type	Sport type of activity. For example - Run, MountainBikeRide, Ride, etc.	3648	object
9	id	The identifier of the activity	3648	object
10	start_date	ISO 8601 formatted date time.	3648	int64
11	start_date_local	The time at which the activity was started in the local timezone.	3648	object
12	timezone	The timezone of the activity	3648	object
13	utc_offset		3648	object
14	location_city		1284	float64
15	location_state		1321	object
16	location_country		3648	object
17	achievement_count	The number of achievements gained during this activity	3648	object
18	kudos_count	The number of kudos given for this activity	3648	int64
19	comment_count	The number of comments for this activity	3648	int64
20	athlete_count		3648	int64
21	photo_count	The number of Instagram photos for this activity	3648	int64
22	trainer	Set to 1 to mark as a trainer activity.	3648	int64
23	commute	Set to 1 to mark as commute.	3648	bool
24	manual	Whether this activity was created manually	3648	bool
25	private		3648	bool
26	visibility		3648	bool
27	flagged	Whether this activity is flagged	3648	object
28	gear_id	Identifier for the gear associated with the activity. ‘none’ clears gear from activity	2871	bool
29	start_latlng	An instance of LatLng.	3648	object
30	end_latlng	An instance of LatLng	3648	object
31	average_speed	The lap's average speed	3648	object
32	max_speed	The activity's max speed, in meters per second	3648	float64
33	average_cadence	The lap's average cadence	2766	float64
34	average_watts	The average wattage of this effort	2819	float64
35	max_watts	Rides with power meter data only	662	float64
36	weighted_average_watts	Similar to Normalized Power. Rides with power meter data only	662	float64
37	kilojoules	The total work done in kilojoules during this activity. Rides only	2819	float64
38	device_watts	Whether the watts are from a power meter, false if estimated	2823	float64
39	has_heartrate		3648	object
40	average_heartrate	The heart heart rate of the athlete during this effort	3458	bool
41	max_heartrate	The maximum heart rate of the athlete during this effort	3458	float64
42	heartrate_opt_out		3648	float64
43	display_hide_heartrate_option		3648	bool
44	elev_high	The activity's highest elevation, in meters	3228	bool
45	elev_low	The activity's lowest elevation, in meters	3228	float64
46	upload_id	The identifier of the upload that resulted in this activity	3644	float64
47	upload_id_str	The unique identifier of the upload in string format	3644	float64
48	external_id	The identifier provided at upload time	3644	object
49	from_accepted_tag		3648	object
50	pr_count		3648	bool
51	total_photo_count	The number of Instagram and Strava photos for this activity	3648	int64
52	has_kudoed		3648	int64
53	athlete.id		3648	bool
54	athlete.resource_state		3648	int64
55	map.id		3648	int64
56	map.summary_polyline		3648	object
57	map.resource_state		3648	object
58	average_temp		2520	int64
59	workout_type	The activity's workout type	461	float64

Implementation Process

Data Collection:

Accessing the Strava data

Full details on how to get your Strava API keys can be found here [Strava Developers]

Protecting API keys.

It's essential to maintain the secrecy of API keys and refrain from including them in public repositories on platforms like GitHub. During the testing phase, I used a credentials file to store the API keys. To ensure their non-disclosure, I added the credentials file to the gitignore file, preventing it from being published on GitHub.

However, for the project to be evaluated by the examiner, access to the API keys is necessary. Therefore, after discussion with the tutor I have chosen to embed the keys directly into the main file while implementing measures to restrict access solely to the examiner. Once the evaluation is complete, I will generate new API keys to ensure the security and integrity of the data.

Extracting Data

The Strava API has a restriction where you can only retrieve a maximum of 200 activities per page. To address this limitation, the following steps were taken:

1. Initially, a variable called "request_page_num" was defined and set to 1.
2. An empty list called "all_activities" was created to store the activities as they are retrieved.
3. A while loop was implemented, with the condition "while true," to continuously fetch the activities and update the "all_activities" list. Additionally, the "request_page_num" was incremented by 1 in each iteration to retrieve the next page of activities.
4. For user convenience, a message displaying the running total of activities downloaded was shown during the process. This allowed the user to track the progress of the data collection.
5. Once all activities were collected, the loop was exited, and the program proceeded to the data cleaning process.

Cleaning data

Data Cleansing

The initial dataset contained unclean data, this was mainly due to a lack of oversight during activity recording. Over the course of 13 years, various devices were used to record the data without sufficient consideration for consistently capturing key metrics. Consequently, certain metrics, such as heartrate and cadence, were sometimes missing. Additionally, an intermittent error has been detected when using the Garmin 905 smartwatch for recording, leading to significant inconsistencies in elevation data for certain activities.

The following data cleansing was implemented.

- Converting activity time from seconds to minutes.
- Converting speed from meters per second (mps) to kilometres per hour (kph).
- Applying pandas date format to all date's columns
- Eliminating small activities.
- Excluding activities that likely occurred in a car.
- Excluding activities with inaccurate elevation information.
- Excluding activities without a heart rate or with a heart rate exceeding 205 beats per minute.
- Excluding outliers, specifically activities lasting over a few days

Regular Expression
To meet the project requirements, we need to incorporate code utilizing Regular Expressions (Regex). Although there is no specific application for Regex in my dataset, I have created an example to demonstrate my understanding of its functionality.

1. Utilize the Google API to perform a reverse lookup on the x and y coordinates from the Start_lating column. Store the retrieved address in a newly created column named "geo_location."
2. Generate a fresh dataset that exclusively comprises activities with the type "Run."
3. Construct a Regex expression designed to identify the occurrence of an "Eircode" within a text string. The expression is as follows: [A-Za-z]{1}[0-9]{2}\s?[0-9AC-FHKNPRTV-Y]{4} Here's a breakdown of the expression:
   [A-Za-z]{1} matches a single alphabetical character.
   [0-9]{2} matches exactly two numerical digits after the alphabetical character.
   \s? allows for an optional whitespace character, permitting a potential space after the two numerical digits.
   [0-9AC-FHKNPRTV-Y]{4} matches precisely four characters from the set of digits (0-9) and uppercase letters, excluding specific letters.
4. Conduct a search within all "Run" activities. If the "geo_location" column contains an "Eircode," extract it and populate a new column called "Eircode."

Data Joining
To fulfil the project requirements, it is necessary to showcase proficiency in joining datasets. However, since there is no additional data available for joining in my project, I have constructed an example to illustrate the understanding of this process.

I proceeded by dividing the original data downloaded from Strava into two distinct datasets. The column named "map_id" served as the shared column between these datasets. Subsequently, I merged the two datasets back together, employing the "map_id" column as the basis for the join operation. The merged dataset was then saved as a CSV file “joined_dataset.csv”

Data Analytics

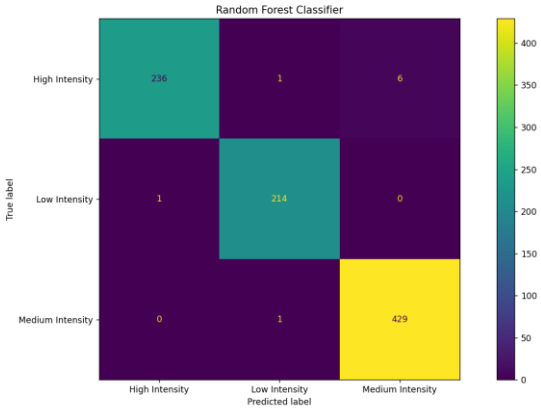
First, I applied KMeans clustering to grade the activities based on several metrics, including moving time, average heart rate, maximum heart rate, average speed, distance, and elapsed time. This clustering process assigned each activity to a specific cluster.

Next, I used the assigned cluster numbers to associate each activity with a corresponding "difficulty label," categorizing them as high, medium, or low intensity.

To validate the accuracy of the assigned difficulty labels, I divided the dataset into a training dataset and a test dataset. Then, I employed the following supervised algorithms:

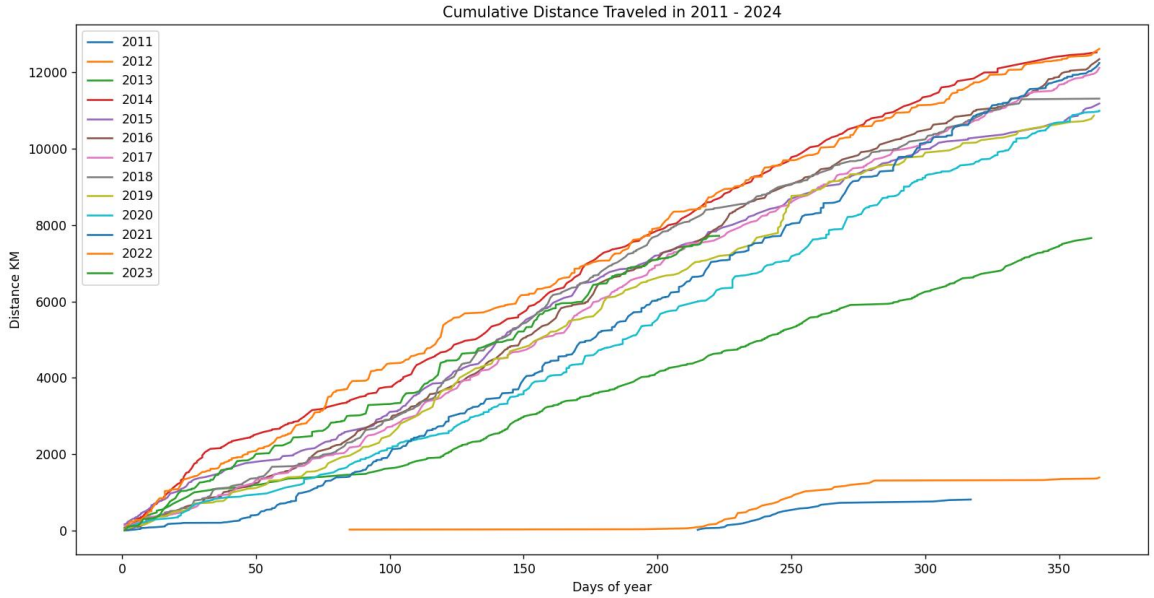
- K-Nearest Neighbours (KNN) with an accuracy rate of 98.2%
- Random Forest with an accuracy rate of 98.65%
- Decision Tree with an accuracy rate of 98.42%

I've provided a visual representation of the confusion matrix for the Random Forest algorithm below.

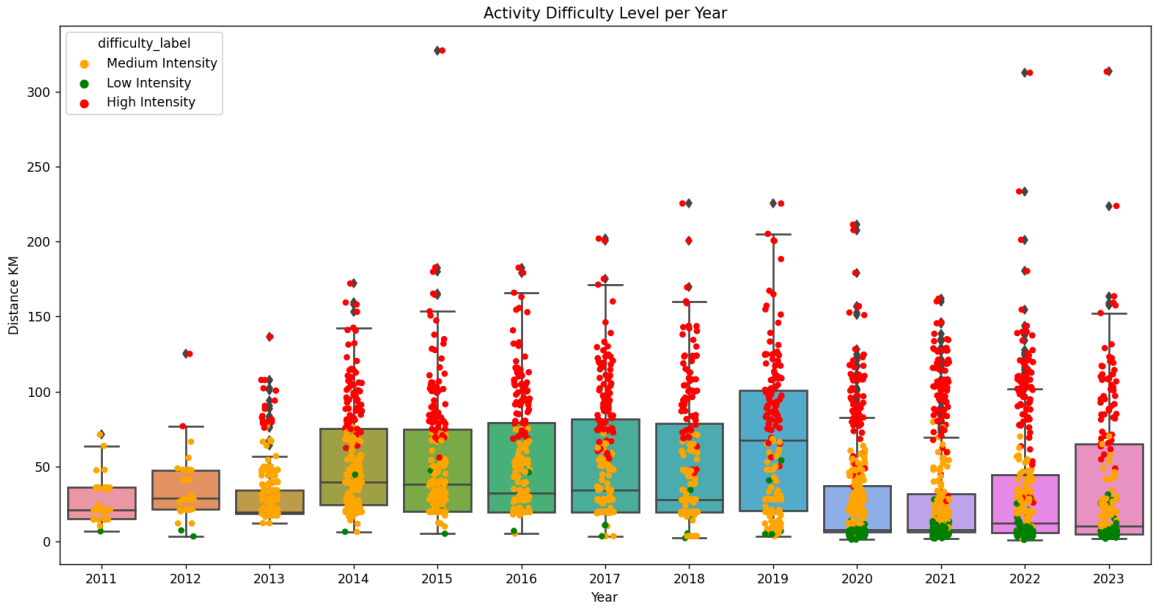


To ensure the availability of all algorithms models for future use, they were saved as a .pkl file, allowing them to be accessed and utilized later if needed.

Results



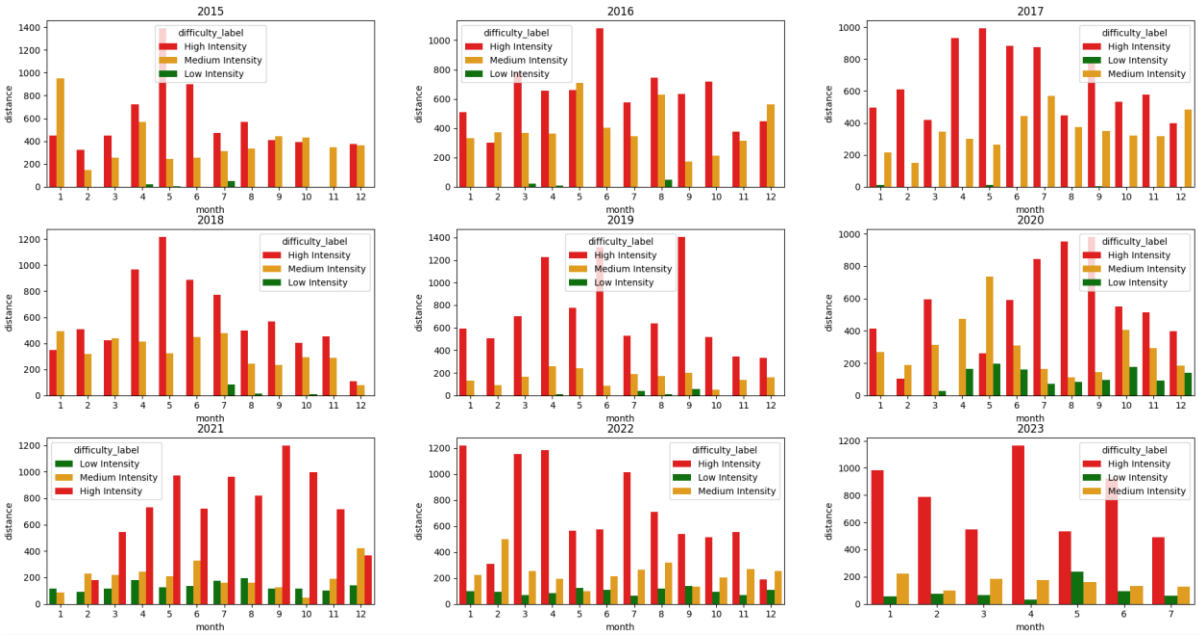
Graph 1 displays a seaborn lineplot graph. It plots the cumulative distance travelled per year broken down by month. Each year is identified by a different colour.



Graph 2 illustrates a Boxplot generated using Seaborn, presenting yearly activity data categorized by distance.

- The box depicted in the plot represents the Interquartile Range (IQR), spanning from the 25th to the 75th percentile. This range encapsulates the middle 50% of the data.
- Inside the box, a horizontal line signifies the Median (Q2/50th percentile), denoting the central value within the data set for that specific year.

Overlaying this Boxplot is a Stripplot, which provides a visual depiction of activity difficulty across each year. This Stripplot distinguishes three distinct difficulty levels: high, medium, and low intensity.



Graph 3 displays the breakdown of activity per month this is useful to identify which months I'm most active and to identify the spread of activity levels throughout the month this can be used to build training plans.

Insights

1. **Annual Cycling Distance Variability**: On average, I cover around 11,500 kilometres per year on my bike. However, it's important to note that the intensity of my cycling varies significantly from year to year. This variability is not immediately apparent without delving into more detailed analysis. January emerges as the crucial month, requiring a concentrated block of training with a minimum cumulative distance of 1000+ Km of high-intensity activities.
2. **Key Importance of January**: One significant observation is that the month of January emerges as a crucial period. It demands a focused block of training, with the goal of accumulating a minimum of 1000+ kilometres through high-intensity activities. This concentrated effort during January is essential for optimal performance.
3. **Opportunity in February and March**: In contrast, the months of February and March show a noticeable dip in intensity. This dip presents a valuable opportunity for substantial improvements. By directing more attention and effort towards training during these two months, I can potentially enhance my overall performance.
4. **Impact of COVID Outbreak**: The outbreak of COVID-19 in 2020 caused a significant shift in my activity patterns. During this time, I engaged in considerable walking, and my cycling activities primarily fell into the 'medium level' intensity category.
5. **2019's Notable Training Change:** A considerable shift in training patterns was evident in 2019 when the average distance covered per activity escalated from approximately 45 km to 75 km

By recognizing these patterns and insights, I'm now better equipped to tailor my training regimen effectively, ensuring that I am adequately prepared for the next Mallorca 213 challenge in 2024.

Conclusions:

The project has been beneficial in identifying training patterns and areas for improvement. However, its effectiveness has been restricted by the manner in which activities were recorded and the limitations of the devices used for recording.

To address these limitations and take the project to the next level, there are plans to collaborate with a semi-professional team to gain access to their data. The team's data offers several advantages over the previous dataset, as it has been collected using top-of-the-range devices such as heart monitors, power meters, and weight trackers, ensuring higher accuracy and control in the data.

It is important to note that the data obtained from the semi-professional team is highly sensitive and private. Therefore, it is imperative to maintain strict confidentiality and ensure that the data is not used for any public project or disclosed to unauthorized parties.

By leveraging this more controlled and accurate data, the aim is to further develop the project and gain valuable insights that could be used to enhance training strategies, optimize performance, and uncover deeper patterns in athletes' activities.

References

Data sourced from: Paul Dardis IE | Strava Cyclist Profile.

Guidance on accessing the Strava API: Strava Developers

Acknowledgment to Trenton McKinney for assisting in resolving an issue related to generating a Python graph: python - How to create a line graph with cumulative distance traveled per year - Stack Overflow

Assistance received for implementing geo-coding through the Google API using: Python code for reverse Geo-coding using Google API - Stack Overflow