Final Year Project Report

Identification of Training Patterns for Race Preparation

Mark Kennedy

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science with Data Science

Supervisor: Dr. Aonghus Lawlor

UCD School of Computer Science
University College Dublin

April 7, 2019

Project Specification

Subject: Time Series Analysis, Machine Learning.

Prerequisites: Machine Learning, Python.

Project Type: Applied Research.

Software Requirements: Python.

Hardware Requirements: No Specific Requirements.

Target Audience: Suitable for both streams.

Preassigned: No.

Background Information: Preparing for a race such as a marathon involves an extensive training period of several months duration. Most runners preparing for such a race follow a training plan which incorporates a mix of short, long and interval runs. In this project we will investigate the training activities of a group of runners preparing for the 2015 London Marathon. The goal of the project is to apply machine learning techniques to identify distinct patterns of the time series activities followed by the runners during their training phase.

Keywords Python, Time-series, Classification, Clustering, Dynamic Time Warping (DTW)

Core: Time-series classification models for training pattern recognition.

Advanced: Build a training target recommender system, based on complete sessions to date and the athletes desired marathon finish time, to help athletes to achieve their desired marathon finish time.

Abstract

Running is undisputedly one of the world's oldest and most accessible sports. Recent years have seen the meteoric rise and advancements in wearable technology (such as fitbits), and fitness apps (such as Strava). These apps allow us to analyse our own individual training sessions in real-time. However, can the data generated from such apps be used for more general analysis of the entire population of athletes?
Sports scientists are always striving to discover the 'optimal' way to prepare for a marathon. Although many of the findings from such studies are infeasible for the large majority of marathoners, due to limited resources available to them, along with other commitments which they may have. This project will attempt to use the data generated by a popular fitness app, to find out how people tend to train for marathons, and the best approach which one could take when doing so.

Acknowledgments

I would like to express a special thanks to Dr. Aonghus Lawlor, for both his time and guidance throughout the duration of this project. His expertise in this area of research helped me to apply many of the techniques learned throughout my studies in UCD into practical applications, while his wealth of knowledge and experience helped me to overcome issues when implementing them.

Table of Contents

1 Introduction 5

1.1 Purpose Of Project 5

1.2 Project Objectives 5

1.3 The Data 6

1.4 Data Preparation 7

1.5 Important Note 7

1.6 Overview Of Process 8

2 Background Research 9

2.1 Time Series Classification 9

2.2 Core Analysis 16

2.3 Advanced 17

3 Findings/Results 18

3.1 Time-Series Classification 18

3.2 Core: Cluster Analysis 22

4 Advanced: Training Targets Recommender 31

4.1 Currently Existing Methods 31

4.2 The Approach 31

4.3 The Output 32

5 Conclusions 34

5.1 Project Findings 34

5.2 Project Review 35

5.3 Possible Future Work 35

5.4 Running The Code 36

Chapter 1: **Introduction**

This chapter will discuss the main motivations/purpose of the project. I will also state the project objectives, which I hope to achieve upon completion of this project. And finally, give a brief oversight of the various approaches and techniques which were adopted over the duration of the project, in order to achieve my goals.

1.1 Purpose Of Project

The main motivation behind this project was to examine data generated by non-elite athletes, as they prepared for a major marathon. The data generated by popular fitness apps/devices is something which isn't being fully utilised currently.
There have been many studies carried out in the past which set out to determine the optimal way to prepare for a marathon. One well known example is Nike's recent experiment called 'Breaking2', which was carried out in 2017. The aim was to complete the first ever marathon in under two hours. The current world record stands at 2:01:39, which was recorded just last year in the Berlin marathon by Eliud Kipchoge of Kenya. Kipchoge also happened to participate in Nike's 'Breaking2' experiment, along with two other elite athletes. During the experiment, the three elites were each given an intensive 6-month training programme, tailored to suit their own personal strengths. The fastest finish time in the experiment was recorded by Kipchoge (2:00:25). Although they failed to achieve their goal, the training programme which they followed is strongly considered to be the most effective to-date.
However, it's not feasible to expect average, non-elite athletes to adopt these training programmes, as they do not have the same amount of resources, spare time or level of fitness as the elites in such experiments. People run marathons for a variety of different reasons, in fact, only a very small proportion of the field actually set out to win. People have their own individual goals, suited to their personal capabilities. The development of wearable technology and fitness apps have opened a new window of opportunity, to analyse how the majority of people train for marathons, and not just the select few elite athletes.

1.2 Project Objectives

In completion of this project, I hope to successfully...

1. Core: Identify the most important feature(s) for an athlete to focus on when training for a marathon.
2. Core: Determine the most suitable way to categorise the different approaches taken by athletes when preparing for a marathon, and to determine which approach is the most efficient.
3. Advanced: Build a recommender model capable of recommending new training targets for individual athletes, based on their completed sessions to-date. These targets will help the athletes to achieve their desired finish time in the eventual marathon race.

1.3 The Data

The dataset was obtained from a popular fitness app known as 'Strava', and contained records of training sessions from 4,487 unique athletes, as they prepared for the 2015 London Marathon.

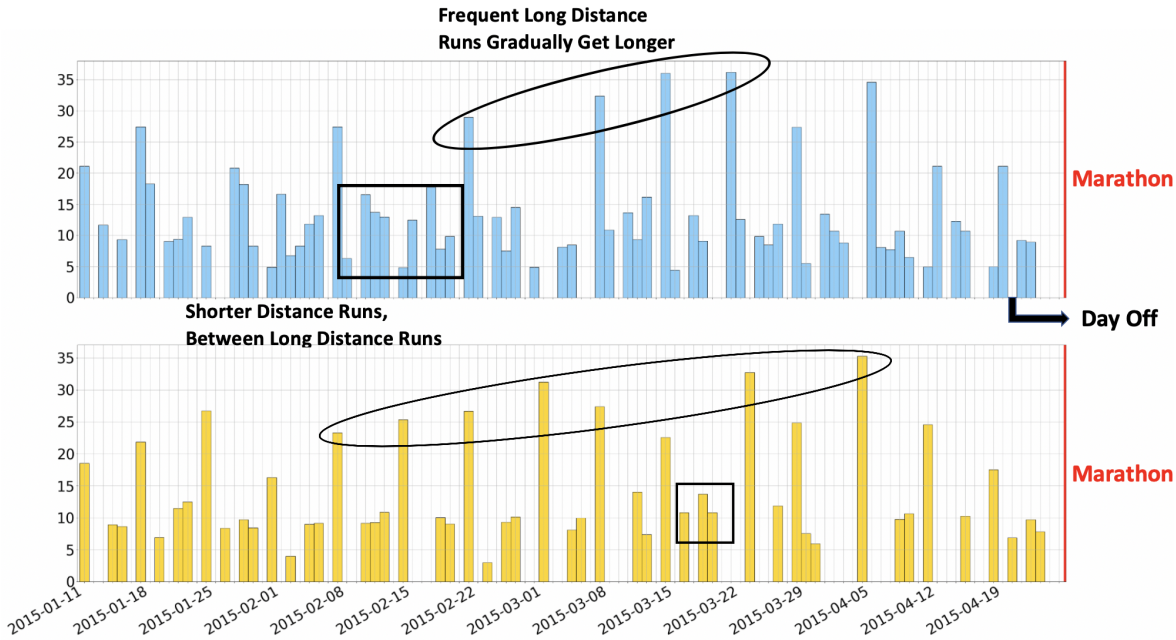

Figure 1.1: Examples of Training Programmes

Figure 1.1 shows an example of two separate training programmes carried out by two different athletes. Each day within the training period is represented by a bar. The height of the bar represents the distance covered by the athlete on that given day (If no bar is present no session was completed). The red bar at the end of the programmes represents the marathon date.

We can clearly see that the two athletes share similarities in their training programmes. They both frequently complete long distance runs (>25km), while completing numerous shorter runs in between. They also increase the distance of the long runs gradually, up until approximately a month prior to the race.

The raw dataset obtained contains 20 columns(features) and over 565,000 rows, which represent distinct training sessions over the course of a year. In figure 1.2 we can see the columns and their respective data types. Notice how the identification numbers of the individual athletes and their training sessions are hashed. This ensures that their identity will remain classified throughout the duration of the project. There is also no sensitive information contained within the dataset, so there are few ethical factors that need to be considered. However, as a precaution, the dataset will not be made public after the completion of the project.

```
hashedid                      object
startdate               datetime64[ns]
startdatelocal          datetime64[ns]
totaldistance                  float64
elevgain                       float64
hashedathleteid                object
startLatApprox                 float64
startLngApprox                 float64
race_id                        object
distance                       object
time                           object
elevation                      object
cadence                        object
heartrate                      object
temperature                    object
pace_diff                      object
pace_cumul                     object
gap                            object
cumulative_elev_gain           float64
cumulative_elev_loss           float64
dtype: object
```

Figure 1.2: Raw Dataset Columns

1.4 Data Preparation

Strava is designed to work on mobile phones and various wearable technologies. It is compatible with both apple and android devices, it targets both runners and cyclists. Care must be taken not to include any individual who may have used a bike or car, resulting in a 'false' session. The most appropriate approach in dealing with such instances was to completely exclude these athletes, as opposed to just removing their suspicious sessions.

As the training programmes were being represented in the form of a time-series, the first task was to merge cases where multiple sessions were recorded by the same athlete on a particular date. The next task was to filter out sessions which were completed outside of the designated training period (the 15 weeks leading up to the marathon).
The original dataset didn't contain the official marathon finish times for the athletes, so an athlete's marathon finish times was determined by finding the element in their time array which corresponded to the 422nd (42.2km) element in their distance array. Once this was determined I filtered out any invalid marathon finish times, such as any time under the eventual winning time.

A combination of existing world records were used to determine suspicious sessions e.g. any session that was recorded over 100 meters travelling at an average speed above Usain Bolt's 2008 world record speed of 37.58km/h was considered to be false, and consequently removed.
These are just some examples of the methods used to cleanse the original dataset. Once this stage of the project was complete, there were 3,311 individual athletes remaining, who together recorded a combined total of 131,230 unique training sessions. Figure 1.3 shows the training sessions which were deemed to be valid.

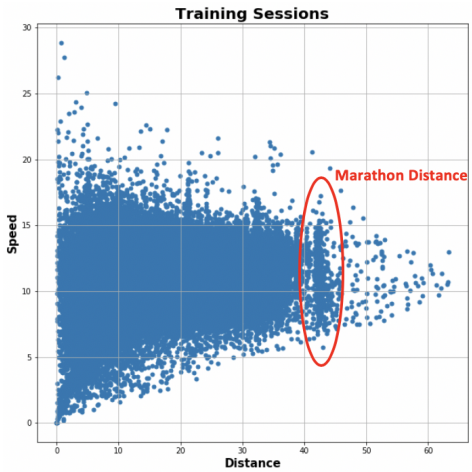

Figure 1.3: The Valid Training Sessions

1.5 Important Note

It is important to understand the difference between pace and speed, before proceeding with the rest of this report.

$$Pace = Time/Distance \tag{1.1}$$

Pace is simply the time it takes to run a set distance. Example: 10 minutes per kilometer. Whereas, speed is the inverse of pace.

$$Speed = Distance/Time \tag{1.2}$$

Speed is the amount of distance you will cover in a set amount of time. Example: 20 kilometers per hour.
What is considered a high pace will also be considered a low speed. And what is considered a low pace, will also be considered a high speed. Speed was used throughout the project as it is easier for people who are not familiar with athletics to relate to.

1.6 Overview Of Process

Over the duration of the project, many different techniques were investigated and experimented with. In the end, some techniques proved more successful than others. A flow chart of the project's overall processes can be seen in figure 1.4. This is included to help envisage the various steps and overall direction of the project. The reasoning behind certain decisions will be discussed throughout the report.

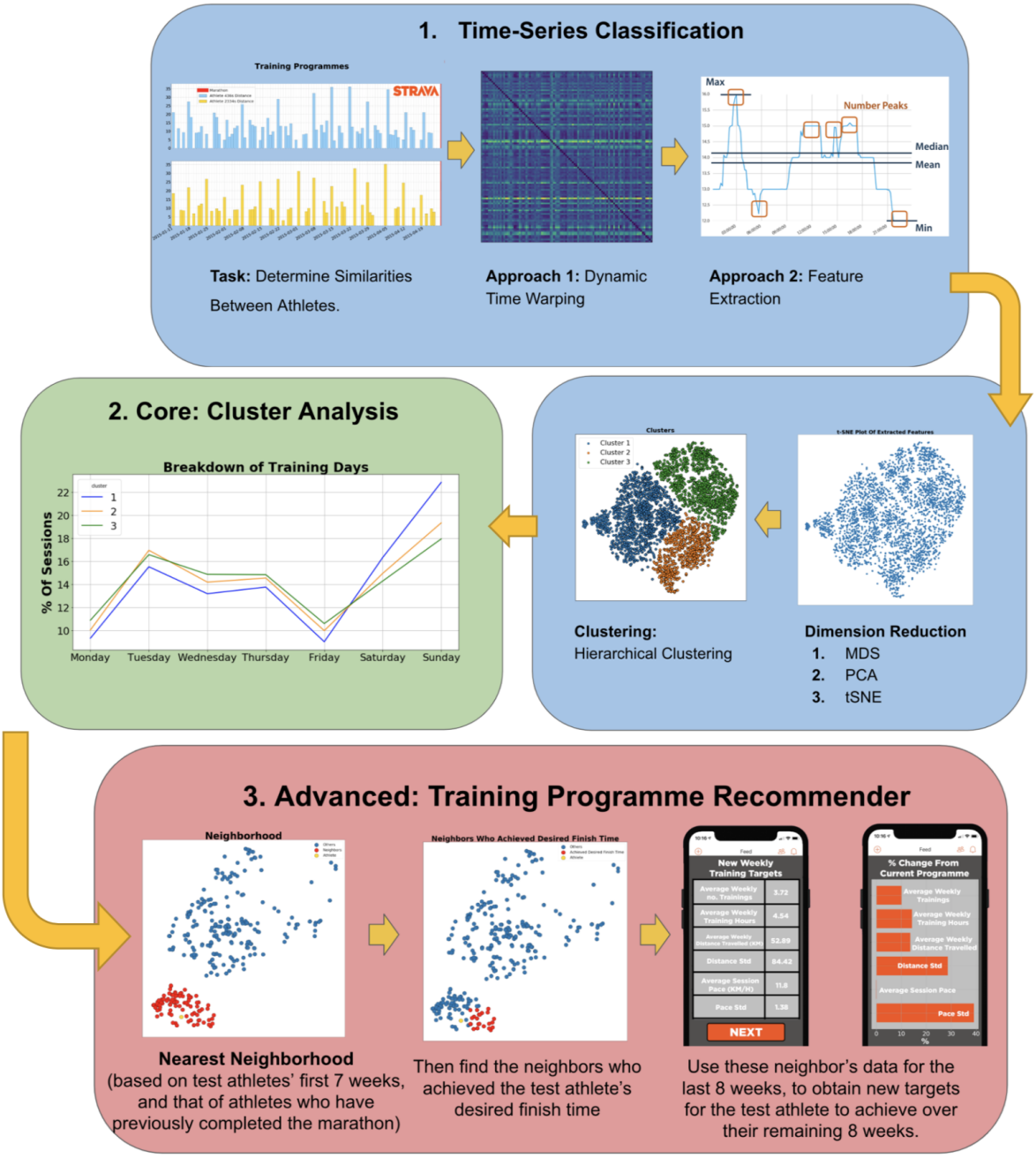

Figure 1.4: Project Process Map

Chapter 2: **Background Research**

In this section, the main methods/techniques applied throughout the project will be discussed stating their advantages and disadvantages along with how they were applied. It is worth noting that not everything mentioned in this section actually ended up being implemented in the end.

2.1 Time Series Classification

The first major task of the project was to group similar athletes together based on their training programmes.
Time-Series Classification is the process of classifying data points based on their behaviour over a period of time. Applying the process to this project will consist of two main steps. The first step is to identify a method of obtaining a 2D plot where each point represents an individual athlete, and the distance between the points reflects their similarity (i.e. the closer they are the higher the similarity between the pair). The second step involves applying an unsupervised clustering technique to the plot obtained from the first step, in order to determine the different clusters of similar athletes.

2.1.1 Approach 1: Similarity Metrics

One approach for the first step is to calculate a metric to indicate just how similar two sequences (training programmes) are. This metric would be calculated based on the values which the time sequences contain at each of the data points. Once a similarity metric for each possible combination of athletes is computed, they will be stored in an nxn matrix, which will be referred to as the 'athlete similarity matrix' throughout this report. The co-ordinates of a given cell, within the matrix, will represent two unique athletes. The cell itself will contain the similarity metric between the two. The resulting athlete similarity matrix will be then passed to a variety of suitable dimension reduction techniques in order to obtain a 2D plot.

Let's consider the two given sequences x and y.

x = [1,2,3,1,2,3,1,2,3,1]
y = [1,1,2,3,1,2,3,1,2,3]

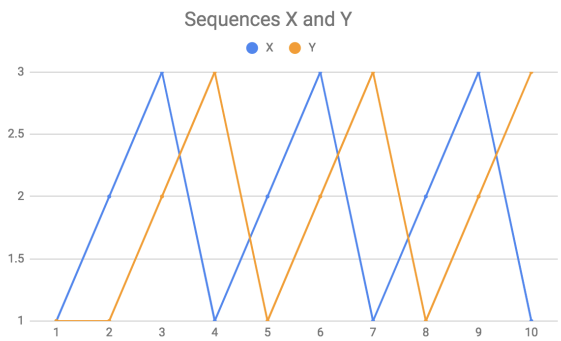

Figure 2.1: X and Y sequences

We can immediately see that they are very similar, apart from the shift in y. What would the most appropriate metric to use in this case?

1. Euclidean Distance

One of the most popular approaches would be to calculate the 'straight-line' distance (between the two sequences) at each corresponding point. This method is known as the 'Euclidean Distance'. The one requirement is that the inputted sequences are of equal length (n).

$$d_{euc}(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

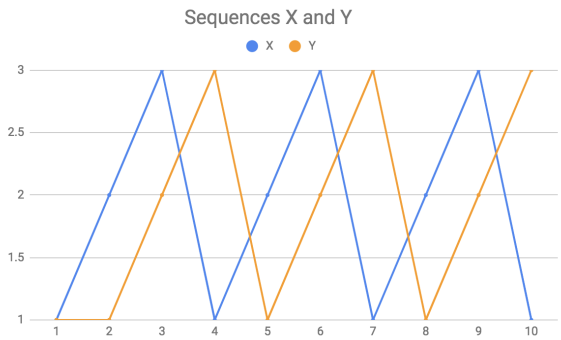

Figure 2.2: Euclidean Distance Applied

As we can see, this is a very simple method to implement and would require very little computational power to execute
However, its major flaw comes in the form of how the distances are recorded. Straight-Line measurements would require two given athletes to carry-out their training sessions on simultaneous days in-order for them to be considered 'similar' by this metric. People generally have busy lives and usually tailor their training programmes around their professional and family lives. As a result, this method is unsuitable for this task.
'The ubiquity of Euclidean distance in the face of increasing evidence of its poor accuracy for classification and clustering is almost certainly due to its ease of implementation and its time and space efficiency.'[1]

2. Dynamic Time Warping

Dynamic Time warping is a method which improves upon Euclidean Distance. It was originally developed to compare different speech patterns in automatic speech recognition. However, "DTW has been successfully applied to automatically cope with time deformations and different speeds associated with time-dependent data"[2]
Unlike Euclidean distance, Dynamic time warping does not require the sequences to be of the same length. However, to ensure that the metric was completely fair to each athlete, it was best to ensure that the sequences were of the same length. And so a 15 week training period leading up to the marathon date was determined as the 'training period', only training sessions which occurred within this time frame were taking into account.

The process involves two steps, the first step is to calculate the 'Cost Matrix'. This is a matrix which contains the absolute distances between each point in the sequences. 'The time cost of building this matrix is O(NM)'[3]

$$c_{i,j} = |x_i - y_j| \tag{2.1}$$

Next we want to find a path p through this matrix that minimises, the total cumulative distance. This path then determines the optimal alignment between the two time-series and is often referred to as the warping path. It should be noted that it is possible for one point in a time series to be mapped to multiple points in the other time series (unlike Euclidean distance which is strictly one-to-one), but the first and last elements in both sequences must be mapped to their counterparts.

$$p = (p_1, p_2, ..., p_k), \text{ where } p_1 = (1, 1), \text{ and } p_k = (N, M) \tag{2.2}$$

The step size condition between elements in the path p, meaning that the step taken between elements in p must satisfy one of these three conditions.

$$p_{l+1} - p_l \in \{(1,1),(1,0)(0,1)\} \tag{2.3}$$

The cost of a given path p is calculated using the cost function.

$$C_p(X,Y) = \sum_{l=1}^{L} c(x_{nl}, y_{ml}) \tag{2.4}$$

Finally, the warping path is defined as the path with the minimal cost that aligns the two sequences.

$$DTW(X,Y) = \min(c_p(X,Y), p \in P^{N \times M}) \tag{2.5}$$

Here is dynamic time warping applied to the x and y sequences.

3	2	1	0	2	1	0	2	1	0	2
2	1	0	1	1	0	1	1	0	1	1
1	0	1	2	0	1	2	0	1	2	0
3	2	1	0	2	1	0	2	1	0	2
2	1	0	1	1	0	1	1	0	1	1
1	0	1	2	0	1	2	0	1	2	0
3	2	1	0	2	1	0	2	1	0	2
2	1	0	1	1	0	1	1	0	1	1
1	0	1	2	0	1	2	0	1	2	0
1	0	1	2	0	1	2	0	1	2	0
	1	2	3	1	2	3	1	2	3	1

Figure 2.3: Cost Matrix Example

3	9	5	6	6	3	3	3	1	0	2
2	7	4	6	5	2	3	1	0	1	2
1	6	4	5	3	2	2	0	1	3	3
3	6	3	3	3	1	0	2	3	3	5
2	4	2	3	1	0	1	2	2	3	4
1	3	2	2	0	1	3	3	4	6	6
3	3	1	0	2	3	3	5	6	6	8
2	1	0	1	2	2	3	4	4	5	6
1	0	1	3	3	4	6	6	7	9	9
1	0	1	3	3	4	6	6	7	9	9
	1	2	3	1	2	3	1	2	3	1

Figure 2.4: Optimal Path

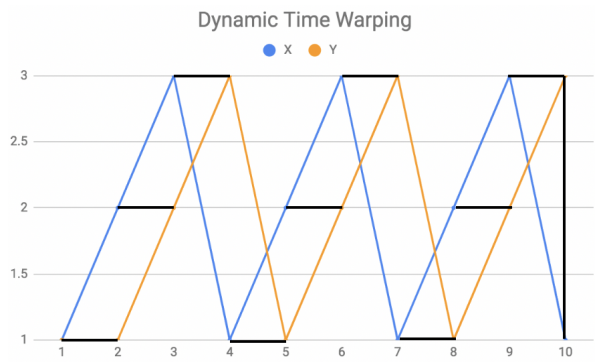

Figure 2.5: Dynamic Time Warping Example

This method will require substantially more computational power and time to implement than euclidean distance. However, its ability to allow for a 'shift' in time between the two sequences, outweighs the disadvantage of it being slower to run. This means that athletes do not have to carry out their sessions on simultaneous days in order for them to be considered similar. Luckily, there is a python module available that computes the metrics accurately and efficiently.

The total time it took to calculate the similarity metrics between 10 athletes, was recorded at 4 seconds (Note: that 10 athletes would return 100 metrics (10x10) to be calculated).
Once the data was cleansed 3,311 athletes remained within in the dataset. Meaning that 10,962,721 similarity metrics will have to be calculated using the dynamic time warping function. Using the result above we can estimate the time it will take to compute these.

$$12s \times (10,962,721/100) = 15days \tag{2.6}$$

The diagonal of the matrix (where the indexes are the same) will be zeros, as the dynamic time warping distance between a time series and itself will be equal to zero.
Also due to the matrix's symmetric nature, only half of the remaining cells will have to be computed i.e. the upper or lower triangular section of the matrix (as the ith ×jth cell will contain the same metric as the jth ×ith cell).

$$(12s \times (3,311 \times 3,310)/100)/2 = 7.5days \tag{2.7}$$

Although this is a major improvement from the first estimate, it is still a significant amount of time to allocate to a single section of the project. Especially if it has to be run multiple times due to possible issues with the results.

Parallel Programming

Parallel programming is a method which could be implemented to further reduce the time required to compute the similarity metrics.
Due to the independent nature of the calculation of similarity metrics (i.e. the computation of one does not affect the computation of another), a parallel programming technique can be applied.
By default one core is used to process the calculations, however, if multiple cores are used, It would speed up the process by n times (n being the number of processors available). Parallel programming is used to break a job down into a series of smaller chunks, it then delivers these chunks along with the instructions to a specified number of core processors. These 'jobs' are executed simultaneously, hence the name 'parallel programming'. In theory, even if only 2 processors were available the time required to execute the function should be approximately half of what it originally was. Using the Secure Shell (ssh) I was granted access to a more powerful computer in UCD, which has 72 cores. I used 60/72 cores as this would leave some cores free for others to use. But even by implementing parallel programming with 60 cores it still drastically improved the wall time of the dynamic time warping process.

$$7.5days/60cores = 3hours \tag{2.8}$$

It is important to remember that this is just an approximate. Some of the metrics may take longer to compute than others due to the complexity of their respective time series.

Once the athlete similarity matrix is obtained, it will then be passed to suitable dimension reduction techniques, which will be discussed later in this chapter.

2.1.2 Approach 2: Feature Extraction

The second approach which was used as an alternative to using similarity metrics was the process of feature extraction. This involves extracting as many features as possible from the time-series, and then passing the resulting high dimensional dataframe, to multiple dimension reduction techniques. 'tsfresh' is a package available for python which is capable of computing hundreds of features from the most simplistic of time sequences. Another advantage of this package is its ability to calculate features on multiple time-sequences for the same id (i.e. total distance, average speed and heart-rate) in a matter of minutes.

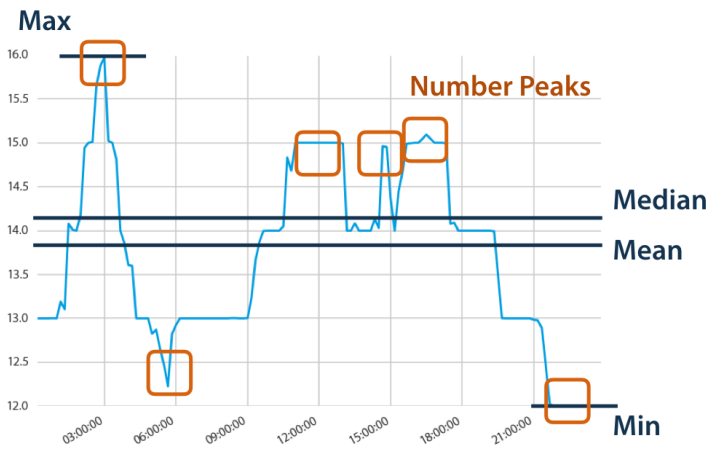

Figure 2.6: tsfresh Package

Some features will be more useful than others, for example, some features might contain null or infinite values, while others may contain the same value for all of the athletes. Such features were filtered out before the dataframe was passed to the various different dimension reduction techniques.

2.1.3 Dimension Reduction

The data obtained up to this point has been high dimensional. When plotting, it is important to covert the data to lower dimensions so that we can interpret the findings easily. It is hard for the human mind to envisage anything above three dimensions, but for the purpose of conveying the results, I will aim to plot 2D graphs. There are many different types of dimension reduction methods.
A suitable method for plotting the athlete similarity matrix generated by the first approach (dynamic time warping) would be Multi-Dimensional Scaling as it has the ability to take pre-computed distances in the form of a matrix as an input.
While the extracted features approach must be plotted differently, as it does not calculate similarity metrics. Luckily most dimension reduction techniques take high dimensional raw data as inputs and compute the differences between the instances automatically (multi-dimensional scaling also has this capability). This means that the newly generated extracted features dataframe can be passed straight to techniques such as Principal Component Analysis and T-distributed Stochastic Neighbour Embedding also. We explore the mentioned techniques below.

1.Multi-Dimensional Scaling (MDS)

Multidimensional scaling is very similar to PCA (it can also be referred to a Principal Co-ordinate Analysis), however, instead of converting correlations into a 2D graph, it converts predefined distances among the samples into a 2D graph (metric scaling).
'In metric scaling the dissimilarities between all objects are known numbers, and they are approximated by distances. Thus objects are mapped into a metric space'.[5]
This means that the true representation of the athlete similarity matrix will be reflected in the plot.
As previously mentioned MDS also has the capability of plotting high dimensional data without pre-computed distances (non-metric MDS). Meaning that it can also be applied to the extracted features dataframe as well as the athlete similarity matrix.

2.Principal Component Analysis (PCA)

PCA is one of the most frequently used methods of dimension reduction, it converts the correlations between the samples (in this case the athletes) into a 2D graph. The axis are called PC1 (x-axis) and PC2 (y-axis). The axis are ranked in order of importance, so differences on the PC1 axis are more important than differences on the PC2 axis. It should be noted that clustering on maximising the linear correlations, is the same as clustering based on minimising the linear distances (Euclidean Distance).
Since the athlete similarity matrix is symmetric, it too can be considered as 'high-dimensional' data, just as the extracted features dataframe is. This means that the technique of PCA could also be applied to it in order to see if it picks up on something which perhaps MDS does not.

3. T-distributed Stochastic Neighbour Embedding (t-SNE):

t-SNE is another form of dimension reduction, that takes a high dimensional dataframe as an input and reduces it to a low dimensional graph, which retains a lot of the original information.

It takes high dimensional data as an input and plots it. It then computes similarity metrics between the points creating a distance matrix. It then randomly projects the points onto a one-dimensional number line, and recomputes the similarity metrics between points creating a new distance matrix. Finally,it moves the points one by one along the line, with the help of a t-distribution (hence where it gets its name from) until the new distance matrix resembles the original one. Similar points attract to one another while dissimilar points repel. The result is a low dimensional graph where the similar points are clustered close together.

In theory, multidimensional scaling is the bested suited method for the first approach. However, the three techniques which have been discussed above are suitable to be applied to both approaches. Luckily scikit-learn[4] provides easy to use implementations of these three dimension reduction techniques in python, which makes implementing all three methods feasible.

So far, we have explored two approaches which could be taken to obtain plots of the athlete. Regardless of the fact that we may see clear and distinct clusters, an appropriate method must be used to determine just how many clusters exist within the plot and to which cluster each individual athlete belongs to. Each cluster should represent a group of athletes that share similar features in their training programmes. Ideally the clusters will be densely populated and spread far apart from each other, however, realistically there will be a few outliers who are between clusters.

2.1.4 Clustering

Unsupervised learning algorithms attempt to identify patterns in the intrinsic characteristics of the data alone, without referring to any class information. One such method is clustering. The goal of clustering 'is not to maximise a utility function, but simply to find similarities in the training data. The assumption is often that the cluster discovered will match reasonably well with an intuitive classification'.[6]

One of the most commonly used approaches is to us K-means clustering, however this method is not best suited for the task, as K-Means requires the number of clusters to be specified prior to running it. This is an issue, as we are uncertain if there are any clusters of similar athletes within the data, to begin with.
Since the athletes in the dataset have no 'class labels', a form of unsupervised clustering will have to be implemented. A suitable method would be to use Hierarchical Clustering.

Hierarchical Clustering

Hierarchical Clustering involves constructing a hierarchy of clusters, in the form of a Dendrogram. The Dendrogram will be based on the plots obtained from the first step.
This approach takes the given plot as an input and begins to form a hierarchy of clusters based on the distances of the data points from one another. This can be approached in a 'Top-Down' or 'Bottom-Up' fashion, depending on how the user wishes.
In the 'Top-Down' approach, all of the data points (in this case athletes), would all start off in the same cluster. The two furthest away clusters would then become partitioned to form two separate clusters. These splits are then performed recursively as one moves down the hierarchy, until eventually, each data point forms its own individual cluster.
Whereas in the 'Bottom-Up' approach, all of the data points, would all start off in their own individual cluster. The two closest two clusters would then join to form one cluster. These merges are then performed recursively as one moves up the hierarchy, until eventually, each data point is in the same cluster.
The next thing to consider is how the distance between two clusters is calculated. This is known as the 'linkage'. There are 3 types of linkages, single, average and complete, each are shown below.

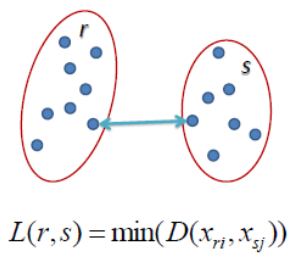

Figure 2.7: Single Linkage

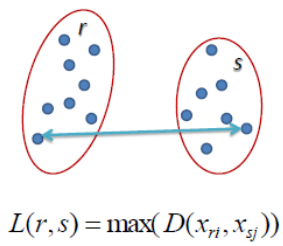

Figure 2.8: Complete Linkage

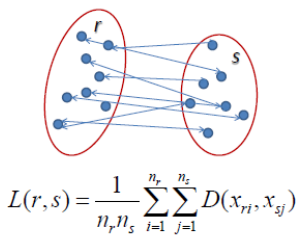

Figure 2.9: Average Linkage

In single linkage (figure 2.7), the distance between any two given clusters is defined as the shortest distance between two points in each cluster. While in complete linkage (figure 2.8), the distance between any two given clusters is defined as the longest distance between two points in each cluster. And finally, in average linkage (figure 2.9), the distance between any two given clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

Once the recursive nature and linkage have been determined a Dendrogram is created to depict the hierarchy of clusters. The user is then required to apply a cut at a depth which they deem fit (depicted as the red line in figure 2.10).
Due to the nature of unsupervised clustering techniques, there is no definitive 'correct' answer. The user may trial many different combinations of linkages and depths, but the 'correctness' of the final product will be determined by their own personal opinion.

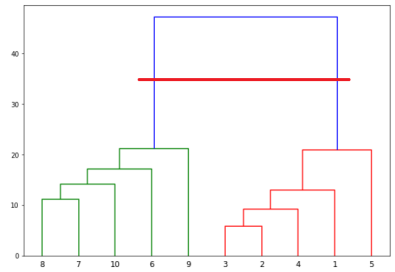

Figure 2.10: Dendrogram Example

2.2 Core Analysis

The next stage of the project is the cluster analysis stage. Once the clusters were obtained, feature selection was used to calculate the importance of the extracted features in determining which cluster each individual athlete belonged to.

2.2.1 Feature Importance

The reason for using feature importance over other methods of feature selection is its simplicity. The feature importance values returned are a direct output of the algorithms used. Other possible methods investigated involved ranking the features by Information Gain.
Feature importance returns a score for each feature within the data, the higher the score more relevant the feature is towards the output variable, which in this case is the cluster label.
Feature importance is an incorporated class that comes with Tree Based Classifiers. Scikit-learn provides an implementation of an Extra-Tree Classifier (extremely randomised trees) which was implemented in this project

2.3 Advanced

The last stage in the project is the advanced stage. The main element in this section will be the implementation of a nearest neighbourhood method. Which will be used to find athletes who have previously completed the marathon, that were in a similar position the test athlete (after n amount of training weeks), when they themselves were training for the marathon.

2.3.1 Nearest Neighbours

K-Nearest Neighbour is a widely used method of classifying test instances based on the instances within the training set which they are closest too (i.e. instances that already have labels).
The k instances within the training set, which the test instance is closest too, are known as the neighbours. Instead of using this technique to classify the test athletes, we will use the neighbour's feature values for the remaining $15 - n$ weeks to set new training targets for the test athlete.

The Nearest Neighbour method determines the 'k' amount of neighbours, by finding the closest 'k' instances, based on the distance between their features and the features of the test instance.
As result the only things that need to be considered are, the value for 'k' (i.e. the number of neighbours), and determining the method for which the 'distance' between instances will be calculated.
The implementation which will be used is available on 'scikit-learn', its default metric for measuring distance is the 'Minkowski' distance.

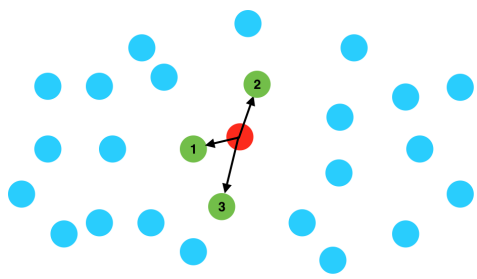

Figure 2.11: kNN Example (k=3)

$$D_{minkowski}(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p} \tag{2.9}$$

When $p = 1$ the Minkowski metric is the same as the Manhattan' distance, and when $p = 2$ the Minkowski metric is equivalent to the 'Euclidean' distance, which I have previously mentioned is also known as the 'straight line distance'. Euclidean distance is the most commonly used measurement, and it is what the measurement which was eventually used in this project.

Chapter 3: **Findings/Results**

This chapter will explore the main findings of the project, discussing how to interpret them, and how these interpretations relate to the project's objectives.
One of the main objectives of this project is to identify how people train for marathons? To answer this we must see if clusters of similar athletes exist within the obtained dataset, and then proceed to examine the main characteristics of each cluster.
Another of the research objectives, is to determine the most beneficial features/patterns for athletes to include in their training programmes when preparing for a marathon? This project will mainly examine the average speed and distance traveled of athletes, and attempt to determine which is most influential for a good marathon finish time.

3.1 Time-Series Classification

The process of Time-series classification was used to determine if clusters existed within the dataset. This section will explore the results of both approaches, and will discuss why feature extraction had to be implemented as an alternative approach to dynamic time warping.

3.1.1 Approach 1: Dynamic Time Warping

The dynamic time warping 'dtw' package was used to compute similarity metrics. These were then stored in the athlete similarity matrix, shown in 3.1. This matrix was then passed to the dimension reduction techniques. Having completed extensive background research on appropriate dimension reduction techniques, it was concluded that Multi-Dimensional Scaling (MDS) was the most appropriate method to use for this approach, primarily because it allowed for pre-computed distances to be used as an input. The athlete similarity matrix was also passed to Principal Component Analysis (PCA) and T-distributed Stochastic Neighbour Embedding (t-SNE). Here are the resulting plots.

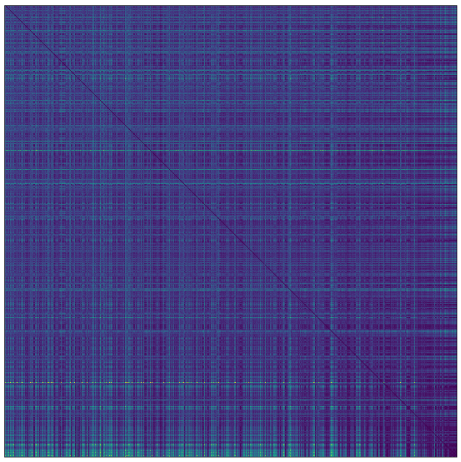

Figure 3.1: Athlete Similarity Matrix

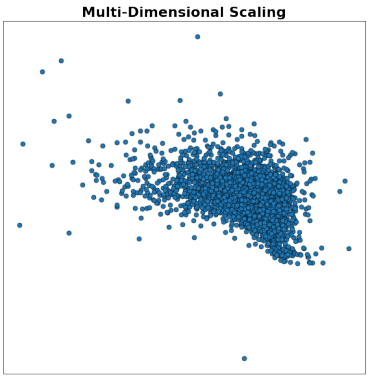

Figure 3.2: MDS

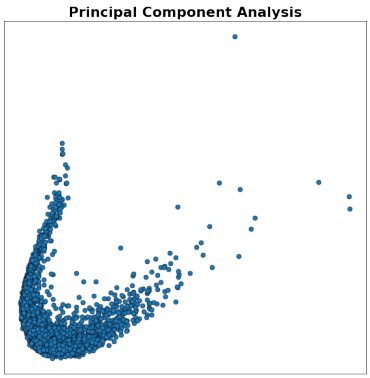

Figure 3.3: PCA

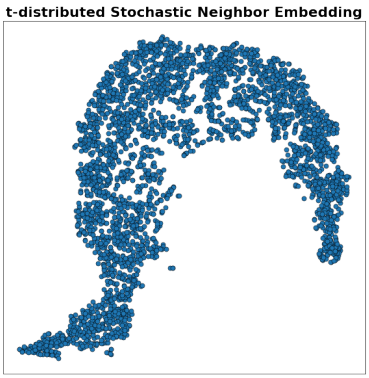

Figure 3.4: t-SNE

Unfortunately, none of the dimension reduction techniques returned a plot consisting of multiple, distinct clusters, as hoped. PCA and t-SNE, both returned interesting plots, but overall hard to interpret and nothing which could be proceeded with. However, the most disappointing result was that of MDS. Instead of returning multiple clusters, the athletes in the MDS plot seem to be concentrated around a fixed spot. This would suggest that all athletes have similar training programmes and that there were no 'sub-groups' of different categories of training programmes.

My initial intuitive was that there exists at least 3 groups of athletes. The first group is athletes who don't run frequently (casually). The second group is those who run frequently but with no real pattern (hobbyists), and finally,those who run frequently and with a set pattern as to when their training sessions occur (elites). The results obtained from this approach contradict my initial intuitive, and my immediate thought was that there was a major flaw with the similarity metrics which had been obtained from the dynamic time warping function.
The next step was to examine the credibility of these metrics. The figure below shows the most similar and most dissimilar training programmes, according to the metrics returned by the dynamic time warping function.

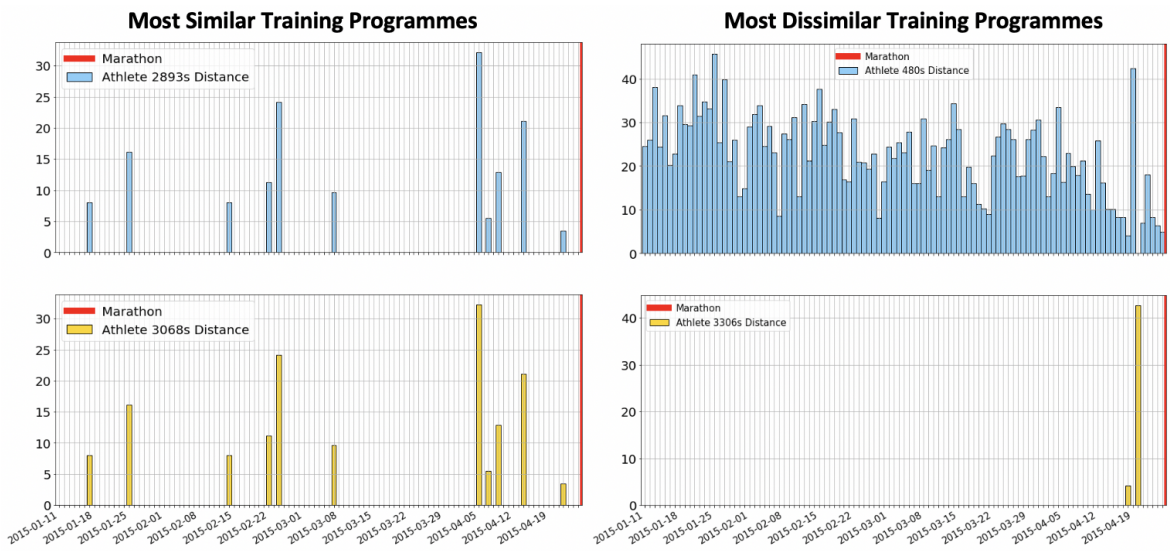

Figure 3.5: Dynamic Time Warping Metrics Examined

Figure 3.5 shows the most similar training programmes (in terms of distance traveled) on the left and the most dissimilar training programmes on the right.
The most similar programmes appear almost identical to each other, in terms of both the distance traveled by the athletes and the dates on which the sessions occurred. This would suggest that they are either running partners or perhaps the same athlete recording their sessions on two separate devices.

The most dissimilar programmes also seem to be plausible, as the top athlete trains on 104 out of the possible 105 training days. And the other athlete trains just twice, with their longest run occurring on the one day that the top athlete does not train.
These are results are beneficial to us, in the sense that they prove that the metrics calculated from the dynamic time warping process are in fact accurate.
This suggests that there are some flaws, to using the dynamic time warping approach for this task.

Possible Flaws With This Approach:

- **The training programmes are too long and noisy:** When beginning this approach I thought it would be sensible to ensure the time series were of the same length, and so I determined a 15 week 'training period' prior to the marathon. Every day within this 15 week period was taken into account, regardless of whether or not the athlete trained on that given day. This led to the time-series consisting of 105 data points (one for each day). This could potentially be too 'noisy' for the dynamic time warping function to return suitable results.
  Possible solutions include shortening the length of the overall training period or taking weekly/monthly aggregates (e.g. sum, mean) instead of the values for the individual days. However, it was decided that it was best to try keep as much level of detail in the time-series as possible before resorting to this approach.

- **Differences in the base levels of athlete's fitness:** This could cause issues in the sense that the time series are on different y-axis scales (distance traveled) as this is dependent on the athlete's fitness. Similar programmes will be determined to be dissimilar if they are on different scales, despite whether or not they are in proportion to one another.
  One possible solution would be to normalise the distance values for each athlete with respect to their own personal min/max values. This would lead to athletes with similar routines being considered 'similar'.
  However just because two athletes share similar routines, shouldn't determine their similarity, as it does not take into account their fitness capabilities.

3.1.2 Approach 2: Feature Extraction

Still confident in finding clusters of similar athletes within the dataset, it was decided to approach the task from a different angle. The second approach involved extracting as many features as possible from the time-series of each athlete, using the 'tsfresh' package. Then passing the resulting extracted features dataframe to the dimension reduction techniques. Both the distance and average speed time-series were passed to the package, in order to base the plot on as much information as possible. Ideally, we would have liked to include even more features such as heart-rate, and cadence (stride rate), however, only a very small minority of the athletes within the dataset, had sufficient recordings for all of these features.
The extracted features which contained 'null' or 'infinite' values for any of the athletes were removed, leaving 368 unique extracted features. These features related to either the athlete's average speed or distance traveled. The resulting dataframe was then passed to the various dimension reduction techniques. Here are the results of this approach.

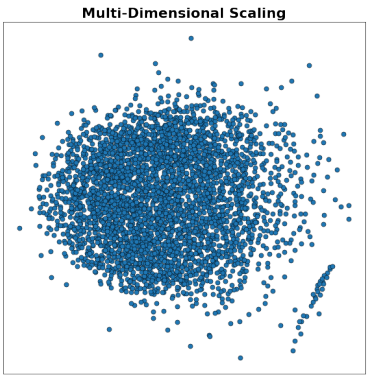

Figure 3.6: MDS

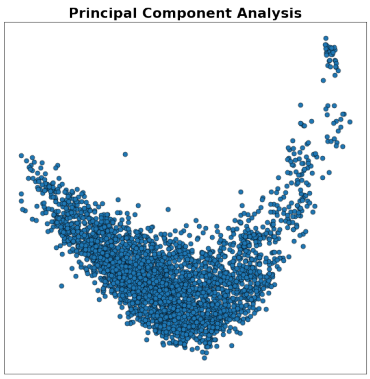

Figure 3.7: PCA

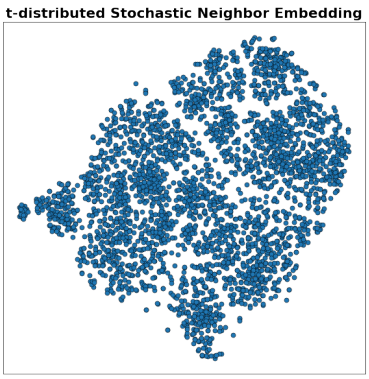

Figure 3.8: t-SNE

In figure 3.6 we can see that the athletes are a lot more evenly distributed than in the MDS plot of the previous approach. However, the athletes are again in one cluster as opposed to multiple distinct clusters. If we look at figure 3.7 we can see the resulting PCA plot of the extracted features. It returned a similar shape to that of the PCA plot in the previous approach, but again unfortunately, nothing which could be proceeded with. Finally,in figure 3.8 we can see the resulting t-SNE plot of the extracted features. Here the athletes seem evenly spread across the plot, with certain spots of high density. This is a plot which could be advanced to the clustering stage.

The first step in the hierarchical clustering method was to construct a hierarchy of clusters in the form of a dendrogram, which can be seen in figure 3.9 (the linkage used in this case was 'average'). The depth at which the cut was applied was determined after experimenting with many different depths and linkages. I felt that this was the most appropriate cut to make in accordance with the plot. The result was three distinct clusters of athletes. If the cut was applied at a lower depth we would have obtained more clusters, but the differences between the clusters in the analysis section would have been minimal.
This brings us to the end of the results for the time-series classification stage of the project.

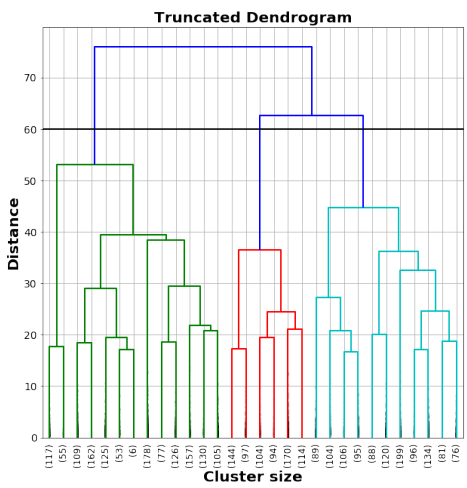

Figure 3.9: Dendrogram

3.2 Core: Cluster Analysis

This section will explore the main findings from analysing the clusters, and explain how they each relate to the research objectives, which were set out at the beginning of this project.
In figure 3.10, we see the plot containing the clusters of athletes. Note the colours of each cluster (1=Blue, 2=Orange, 3=Green), as the clusters will primarily be referred to by their colour throughout this section.

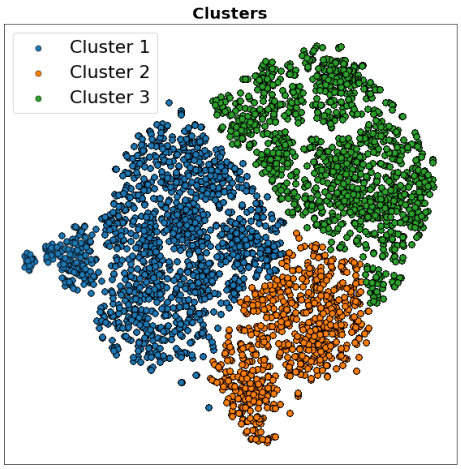

Figure 3.10: Clusters

3.2.1 Finish Times

The first thing which was examined was the performance of each cluster in the actual marathon itself, in order to give us an indication of how the clusters compare against each other in terms of ability levels.

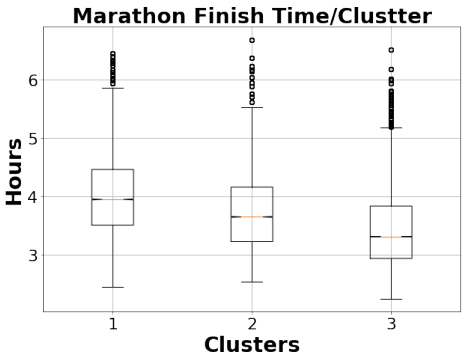

Figure 3.11: Marathon Finish Time Boxplot

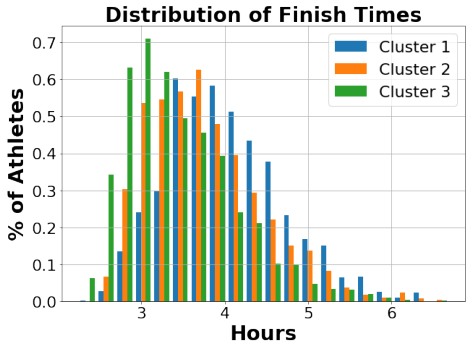

Figure 3.12: Marathon Finish Time Histogram

What do these graphs show?
As we can see in figure 3.11, there is a gradual increase in the average marathon finish time of each cluster (represented by the orange line inside the boxes). Cluster 3 has the worst (highest) average finish time, followed by cluster 2, while cluster 1 has the best (lowest) average finish time. The box plot also highlights the fact that there are outliers present in each of the clusters. It is also worth noting, that the quickest finish time in cluster 1 was better than the quickest finish time in cluster 2, in the actual marathon.
The general ranking of clusters is also conveyed in the histogram shown in figure 3.12 as we can see that green cluster is slightly more shifted towards the left, followed by the orange cluster, and finally,the blue cluster is depicted as the worst performing, as it is the furthest shifted towards the right.

What findings can we take from these graphs?

- The main thing to take from these graphs is the ranking of the clusters, in terms of performance in the marathon. The faster athletes tend to be in the green cluster, while the slowest athletes seem to be in the blue cluster, with the orange cluster serving as an intermediate between the two.

It is also encouraging to see such a shift in the performance of the clusters in the marathon, considering that the time-series classification process only took into account their training sessions prior to the marathon date.

3.2.2 Training Sessions

The next thing which was investigated was the extent to which the athletes within each cluster spent training, over the designated training period leading up to the marathon.

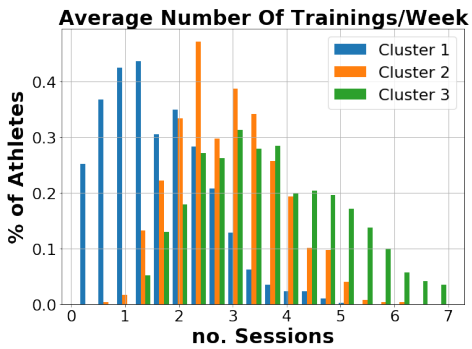

Figure 3.13: Avg no.Sessions/Week

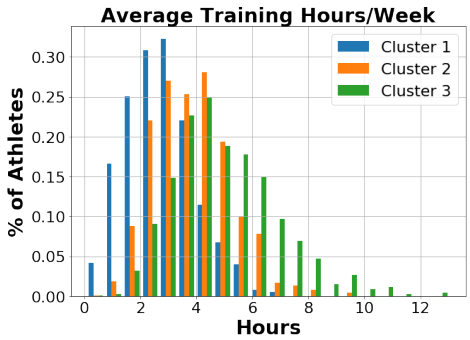

Figure 3.14: Avg Hours/Week

What do these graphs show?
We can see an even bigger shift between the clusters here. Especially in figure 3.13 which shows a histogram of the average number of training sessions per week, by the athletes in each cluster. The athletes in the blue cluster clearly tend to train the least, while the athletes in the green cluster tend to train the most, with the orange cluster again acting as an intermediate between the two.
Figure 3.14, which is a histogram of the average hours spent training by the athletes of each cluster per week, reflects the same portrayal of the clusters as figure 3.13 does.

What findings can we take from these graphs?

- If we combine these findings with what we already know, we can conclude that the green cluster trains the most but also performs the best on the day, and the blue cluster trains the least but also performs the worst in the race. The orange cluster is between the two in terms of both the amount of training they do and how they perform in the actual race.

- Also, quite a few athletes in the blue cluster averaged less than one training session per week. This could explain why it contained outliers, perhaps some of the athlete's didn't record all of their sessions throughout the training period (such as the fastest athlete in cluster 1).

Now knowing the extent to which the athletes in each cluster spend training, the next thing which was examined was the breakdown of exactly when the training sessions in each cluster occurred.

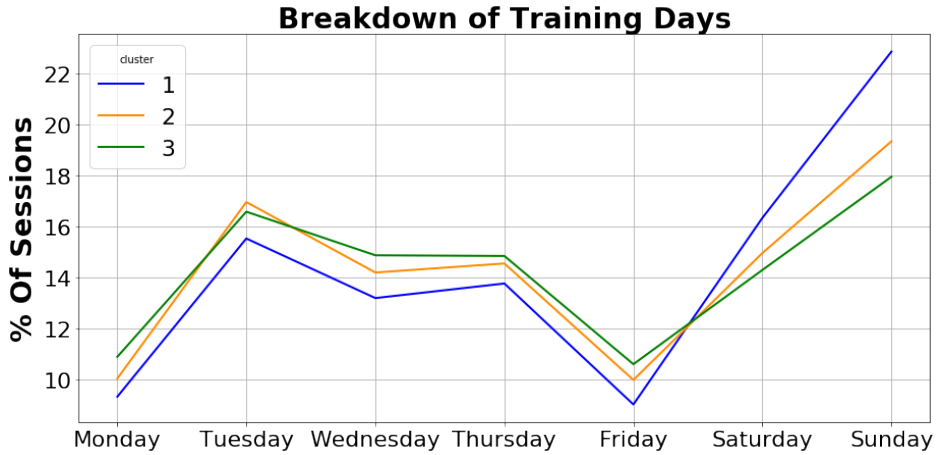

Figure 3.15: Breakdown Of Weekdays

What does this graph show?
Figure 3.15 shows the percentage of training sessions in each cluster, that occurred on each given day of the week.
The most obvious feature in the graph is the spike in the number of training sessions which occur over the weekend, especially in the blue cluster. The two strongest performing clusters in the marathon (orange and green) train more throughout the working week, than their blue counterparts.

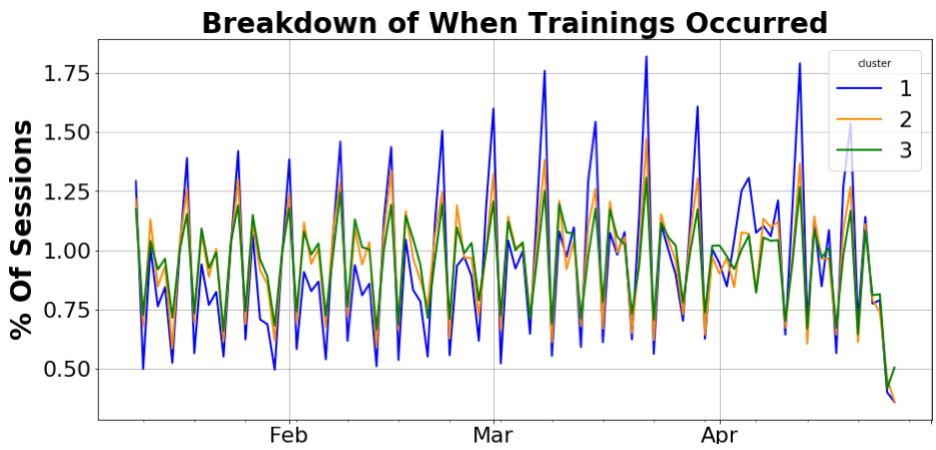

Figure 3.16: Breakdown Of Dates

What does this graph show?
Figure 3.16 shows the percentage of training sessions in each cluster, that occurred on each given date over the training period.
We can see the spike in the number of training sessions occurring over the weekends. Also notice how the blue cluster begins to match (and even out train their counterparts) with about a month to go until the race.
There is also general trends which the clusters seem to share, such as the steady growth in the number of sessions occurring, up until approximately a month prior to the marathon, when the number of sessions occurring seems to peak.
Finally,all three clusters experience a pivot within the final week leading up to the race.

What findings can we take from these graphs?

- One obvious finding is that the weakest performing cluster in the marathon (blue) predominantly trains on the weekends, while the stronger performing clusters train more consistently throughout the working week. This would suggest that they are more committed to the sport and are not just casual runners, like their blue counterparts.
- The late growth in the number of training sessions in the last month by athletes in the blue cluster, would suggest that they leave it too late before they start to train regularly, and subsequently perform the worst in the marathon.
- Another possible interpretation of this graph is that the bulk of the training sessions are completed roughly a month prior to the marathon race. As the number of sessions that occur in the final month seems to generally decline.
- Finally, all of the clusters experience a pivot in the final few days leading up to the marathon, suggesting that the athletes in all of the clusters begin to rest themselves for the race.

3.2.3 Distance and Speed

The next obvious area of research for to look at the main features (distance and speed) of the athletes within each cluster.

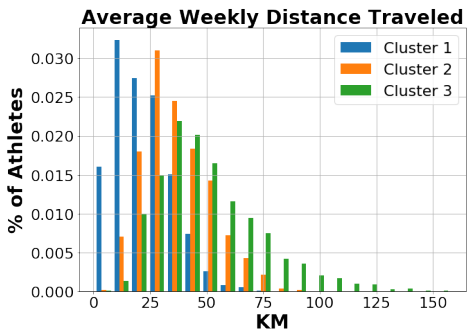

Figure 3.17: Avg Distance/Week

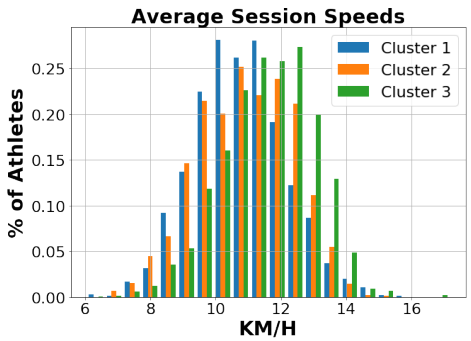

Figure 3.18: Avg Session Speed

What do these graphs show?
Figure 3.17 shows a histogram of the average weekly distance traveled by the athletes. We see that the general ranking of the clusters is once again reflected here, with the athletes in the green cluster covering the most ground per week. While athletes in the blue cluster tend to cover the least, and athletes in the orange cluster covering an intermediate distance between the two. Figure 3.18 shows a histogram of the average session speed of athletes over the training period, again reflecting the same ranking of the clusters with the green cluster containing the fastest athletes in training, and the blue cluster containing the slowest athletes in training.

What findings can we take from these graphs?

- It's unsurprising to find that the athletes in the green cluster cover the most ground weekly, as we have just previously discovered that they tend to train the most (and also for the longest periods of time).
- The main thing which we can take from these graphs is that figure 3.18, confirms that the fastest cluster of athletes in the marathon (green), are in fact the fastest cluster of athletes over the entire training period. And that the slowest cluster of athletes in the marathon (blue), are in fact the slowest cluster of athletes over the training period.

These findings are unsurprising but I'm intrigued to know which feature is most important for athletes when training for a marathon. Is it how much ground they cover? Or is it how fast they run?
To exam this we will have to find the correlations between both features (distance and speed), and the athlete's marathon finish time.
Note: Correlations such as these are known as 'negative correlations', in other words, one variable increases (distance/speed) as the other decreases (marathon finish time).

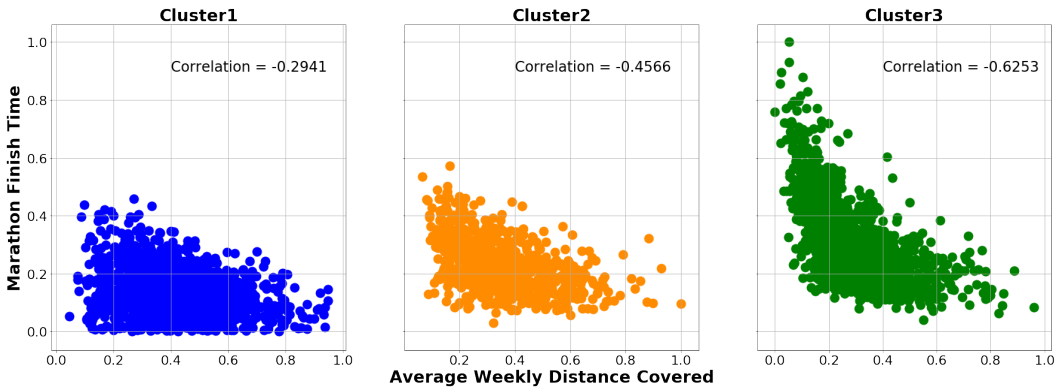

Figure 3.19: Average Weekly Distance vs Marathon Finish Time

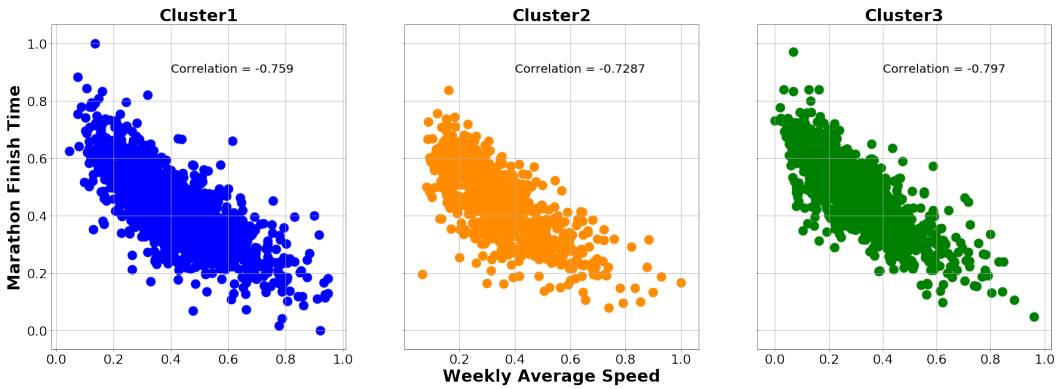

Figure 3.20: Average Session Speed vs Marathon Finish Time

What do these graphs show?
These are very intriguing graphs, as figure 3.19 shows that there is no real correlation between a high average weekly distance traveled and low marathon finish time unless you are an athlete in the green (superior) cluster.
However, if we look at figure 3.20 we can see strong correlations between a fast average session speed and a low marathon finish time, regardless of what cluster you're in.

What findings can we take from these graphs?

- Speed is clearly the most important feature for an athlete to focus on, regardless of which cluster they are in. Athletes constantly fixate about the distance which they are covering in training, when instead they should be focusing on increasing their speed.

Average speed and distance traveled are features which were available in the original dataset, however having plotted the athletes based on the 368 extracted features, we should examine some of these features to see which ones were most significant in determining which cluster each athlete belonged to.

3.2.4 Extracted Features

To do this a feature selection approach (scikit-learn's implementation of Extra-Tree Classifier) was used to calculate the importance of each feature. The results can be seen below.

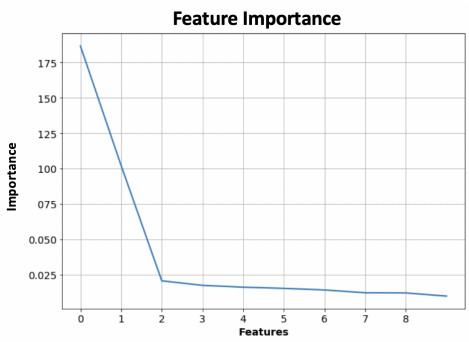

Figure 3.21: Feature Importance 1

Feature	Importance
session_speed__large_standard_deviation__r_0.2	0.186727
totaldistance__large_standard_deviation__r_0.25	0.102147
session_speed__ratio_beyond_r_sigma__r_3	0.020503
session_speed__large_standard_deviation__r_0.25	0.017303
session_speed__approximate_entropy__m_2__r_0.3	0.016004
totaldistance__ratio_beyond_r_sigma__r_3	0.015167
session_speed__binned_entropy__max_bins_10	0.014042
totaldistance__symmetry_looking__r_0.1	0.012038
session_speed__symmetry_looking__r_0.05	0.011920
session_speed__fft_coefficient__coeff_0__attr_"abs"	0.009689

Figure 3.22: Feature Importance 2

What do these graphs show?
As we can see from the figures above the top two features accounted for almost thirty percent of the total feature importance. And these two features are also very similar to each other, other than the fact that one is based on speed, while the other is based on distance.

What can we take from these graphs?

- We can see that the feature based on speed has almost twice the importance of its distance counterpart, strengthening the proposition that an individual's average speed is more important than the distance their covering.

Let's examine the two most important extracted features in more detail. Here is the definition of the features taken from the 'tsfresh' website.

Large Standard Deviation(X, R): A "Boolean variable denoting if the standard deviation of x is higher than r times the range = difference between max and min of x. Hence it checks if

$$std(x) > r * (max(X) - min(X)) \tag{3.1}$$

According to a rule of the thumb, the standard deviation should be a fourth of the range of the values."

But how can we interpret these features?
Let's just consider that '1' represents athletes with a large standard deviation in speed or distance, and '0' represents athletes with a small standard deviation, disregarding the thresholds for the time being.

Then what exactly does it mean for an athlete to have a small standard deviation?
Well if we consider an athlete who generally runs according to a set training routine (based around distance), the distances they cover in training will be quiet similar, meaning that they will have a 'small' standard deviation. Whereas a runner who runs without following a set training routine will generally have a wide variety of distances contained in their training programme, meaning they will have a large standard. The same can be applied for speed.

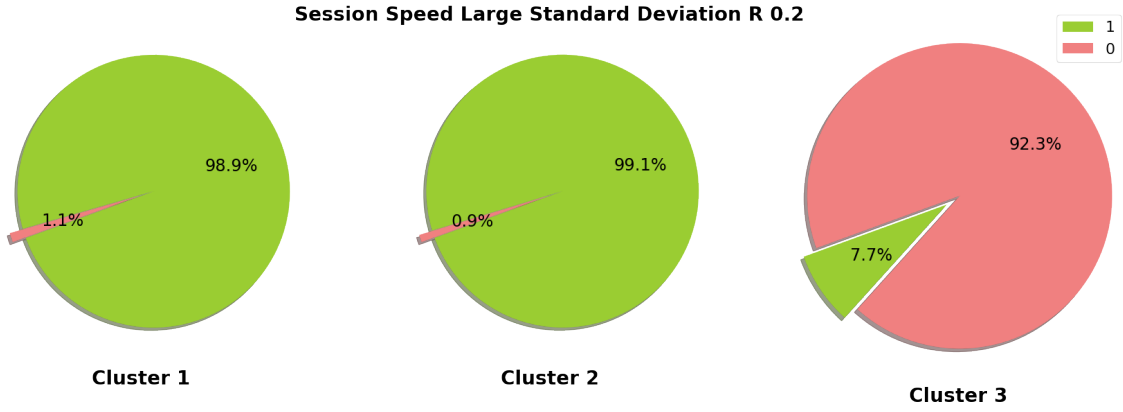

Figure 3.23: Extracted Feature 1: Speed

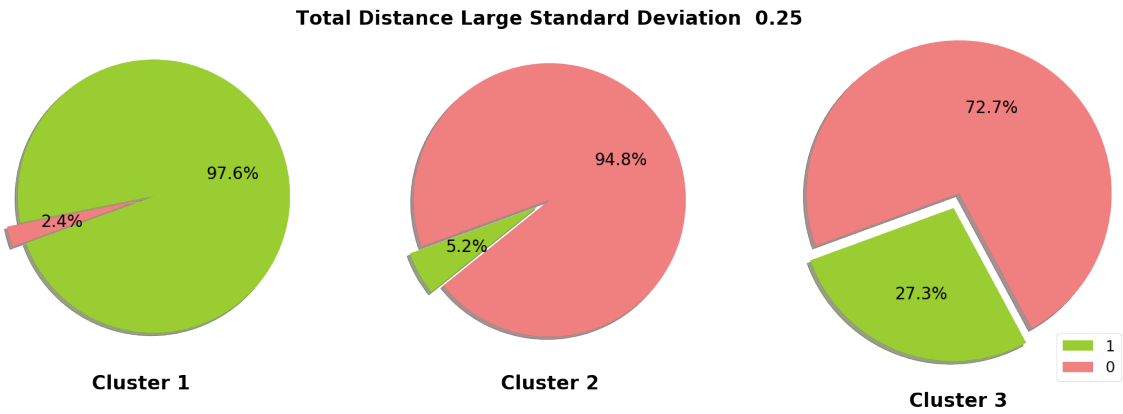

Figure 3.24: Extracted Feature 2: Distance traveled

What do these graphs show?
We can see here that each cluster generally has different combinations of the two Boolean variables.
For example, the athletes in the first cluster (worst performing) mainly have large standard deviations in both average speed and distance traveled.
While athletes in cluster 2 have a large standard deviation for average speed but a small standard deviation for distance traveled.
Finally,the athletes in the third cluster (best performing cluster) mainly have small standard deviations in both average speed and distance traveled.

What findings can we take from these graphs?

- Once we know exactly what it means to have a large or small standard deviation, these graphs become very interesting and important in the grand scheme of the project.
- Athletes in cluster 1 (worst performing cluster), generally have a large standard deviation in both speed and distance, suggesting that they tend not to follow any set training programmes.
- Whereas athletes in cluster 2 have a small standard deviation for distance traveled and a large standard deviation for speed, suggesting that they tend to follow a set training plan focused primarily on distance.
- Finally,the athletes in cluster 3 (best performing cluster), generally have small standard deviations in both speed and distance, suggesting that they are committed to training programmes that are focused on both speed and distance.

There also appears to be quiet a few athletes within this cluster who follow training programmes focused primarily on speed alone (i.e. have large standard deviations for distance). Could this mean that following a training programme based primarily on speed alone, is better than following a programme based solely on distance? One thing which is certain is that the athletes in the third cluster understand the importance of speed.

3.2.5 Training Speed vs Marathon Speed

So far in this section, we have discovered the huge importance of speed. But when training for a marathon how fast do the athletes in each of the clusters train with respect to their actual marathon speed?
There are countless theories available online regarding how fast you should train with respect to your eventual marathon speed. A study currently being carried out by people in the 'Insight Centre for Data Analytics' in UCD, Ireland. Found that "faster marathoners train, on average, at higher paces than their eventual race speed".[7]
It's important to remember that pace is the inverse of speed. So a high pace refers to a slow speed. We can now derive from the quote above, that the faster athletes in the marathon tend to train slower than their actual marathon speed.

Let's examine the percentage of athletes within each cluster who actually train below their marathon speed.

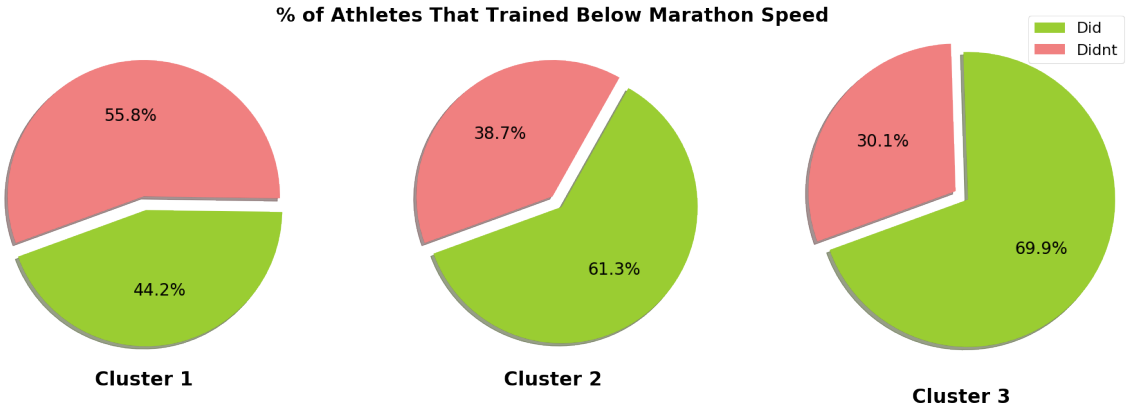

Figure 3.25: Average Training Speed vs Marathon Speed

What does this graph show?
We can see here that the athletes in the better performing clusters do infact train at a slower speed to their eventual marathon speed. And so the findings of the study carried out by the 'Insight Centre for Data Analytics' in UCD, are true for this dataset too.
Cluster 1 (weakest performing cluster) is the only cluster in which the majority of athletes train at a faster average speed over the training period, compared to their eventual marathon speed.

What findings can we take from this graph?

- Now knowing the importance of speed, and the fact that the athletes in cluster 3 tend to follow training plans focused on speed. I was surprised by the high level of athletes within clusters 3, who actually train below their average marathon speed. Perhaps this suggests that they have confidence in their training programmes and are aware of their capabilities, so when the marathon comes around they are more prepared than the athletes in cluster 1.
- High levels of athletes in cluster 1 have a higher training speed compared to their eventual race speed. I suspect that this is primarily because the athletes in cluster 1 are just casual athletes and lack real sufficient experience of long distance runs, and so when it comes to the marathon itself they haven't developed appropriate pacing plans.

Chapter 4: **Advanced: Training Targets Recommender**

This chapter will discuss the advanced stage of the project. The task was to create a training target recommender system which is capable of producing accurate weekly training targets for test athletes, based on both their completed sessions to date and their desired marathon finish time.

4.1 Currently Existing Methods

Current approaches for receiving advice on what targets you should be aiming for when training for a marathon include, going to a professional coach, or using an online calculator. However, it's hard not to be sceptical of the credibility of these approaches.
The targets received from a coach will primarily be based on what they have seen you do so far in training and their opinion on what you should do next. While most online calculators available, take your most recent, significant training session along with your desired marathon finish time as inputs, and calculate your new training targets using a hidden formula. Can a more trustworthy and accurate approach be created, using the obtained data?

4.2 The Approach

Due to the fact that marathons are always the same distance, and that they usually maintain the same course as previous years, it is fair to say that the same race is being run every year (apart from the weather). This means that data obtained from previous years will be relevant for future years (assuming that the course stays the same).
By taking a test athlete's completed sessions in the first n weeks to-date, and their desired marathon finish time as an inputs (example shown in figure 4.1), we should be able to calculate new training targets for the test athlete. This can be achieved by finding athletes who were in a similar position to the test athlete and subsequently went on to achieve the desired finish time when they themselves completed the marathon.

Figure 4.1: Input Example

These new targets will give the test athlete guidance on what they should be aiming for over the last $15 - n$ weeks, if they are to achieve their desired finish time. To do this we will need to implement a nearest neighbourhood approach, based on athletes who have previously completed the marathon.

The first step (figure 4.2) was to find a nearest neighbourhood of athletes who have previously completed the marathon. The nearest 'k' neighbours, will be athletes who were in a similar position to the test athlete, when they themselves were training for the marathon (after n weeks). The value of 'k' was set to ten percent of the total population of the training set.
The second step (figure 4.3) was to find the subgroup of nearest neighbours who achieved the test athlete's desired marathon finish time, when they themselves completed the marathon.
Once the subgroup was found the final step involved finding the average feature values of the subgroup over the last $15 - n$ weeks. These averages will then serve as new training targets for the test athlete to achieve over the last $15 - n$ remaining weeks.

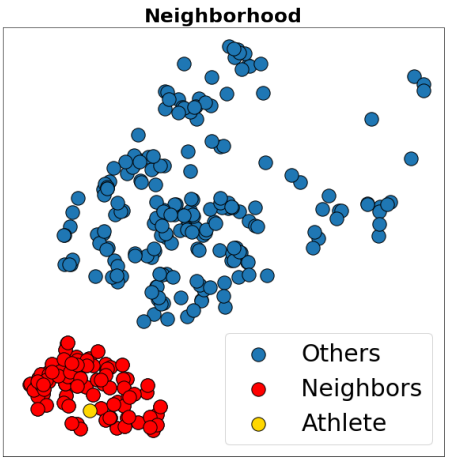

Figure 4.2: Step 1

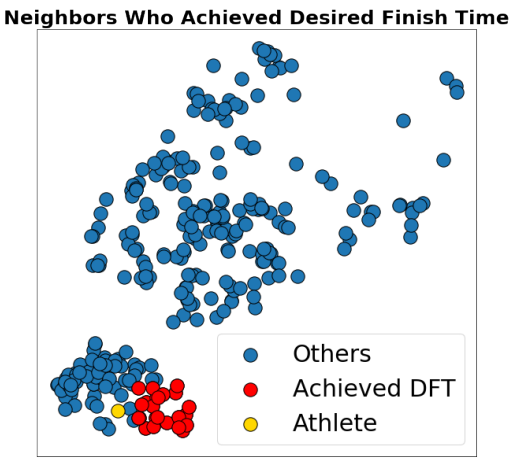

Figure 4.3: Step 2

4.3 The Output

Figure 4.4 shows an example of training targets generated for a random test athlete. Although the nearest neighbours were obtained using the extracted features, It was best to base the new training targets on the basic original features, primarily because they are easier for the user to interpret.
The standard deviation metrics were included for both speed and distance, as we saw their importance in the feature selection stage of the cluster analysis section. The user can interpret this metric as an indication of variety in speed/distance which they should be achieving in their training programme.


```
{
    "New Targets":{
        "average_weekly_trainings":4.3129088129,
        "average_weekly_training_hours":5.2184243483,
        "average_weekly_totaldistance":64.1185538153,
        "totaldistance_std":8.2956389671,
        "session_speed":12.476166148,
        "speed_std":1.4911355777}
}
{
    "Percentage Change":{
        "average_weekly_trainings":43.7636270981,
        "average_weekly_training_hours":40.2155592208,
        "average_weekly_totaldistance":62.9088575018,
        "totaldistance_std":32.642664404,
        "session_speed":16.6660936203,
        "speed_std":50.2351231693}
}
```


Figure 4.4: Example Output

Using these new training targets, along with the test athlete's current feature values, the percentage change between the two was generated to allow the user to quickly identify the aspects of their current programme which they need to prioritise.
Figure 4.4 is a prototype for an app interface, which could work alongside the current Strava app. The first screen displays the users new training targets, while the second screen shows a bar chart representing the percentage change of the new targets from their current programme, for easier interpretation.

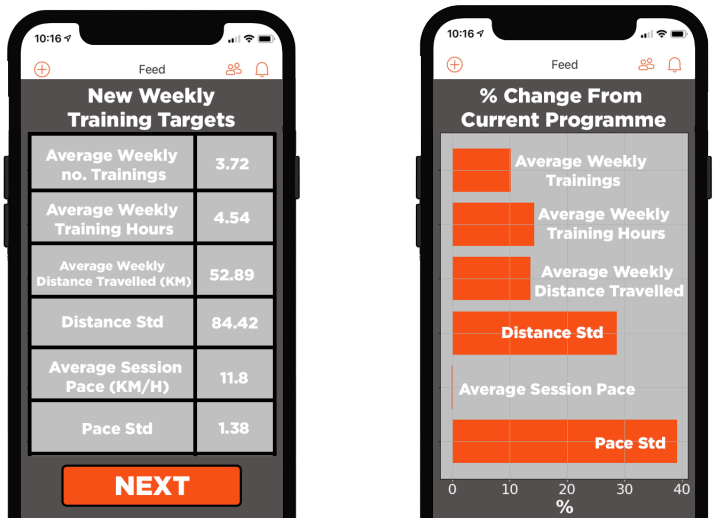

Figure 4.5: Prototype For App

I believe there is huge potential for such a recommender system because, as the years go by, more and more athletes will complete the marathon, meaning that the recommender's targets will become increasingly more accurate, helping more and more individuals achieve their desired finish times. If similar data was obtained on other marathons (e.g. Berlin, Boston), the recommender model could easily be adapted to work for the different marathons. Perhaps it could even be adapted to recommend marathons which the user might fare better in, based on their training sessions to date.

Chapter 5: **Conclusions**

This section will reflect on the findings which were obtained over the duration of the project, and how they relate to the original research objectives.
I will also review the project process as a whole, discussing what I thought went well with the project and also the elements of the project which I would change if a starting again.
Finally,I will also discuss some possible avenues for further work in the future.

5.1 Project Findings

The first research objective was to identify the most important feature(s) for an athlete to focus on when training for a marathon.
When considering all of the results obtained throughout the analysis section, there can only be one answer to this, and that is 'Speed'. We saw with the correlation graphs that speed is the number one most important thing for all athletes, regardless of what cluster they are in. The majority of inexperienced athletes constantly fixate about the distance which they are covering, when in fact they should be more focused on the speed at which they are running.

The second research objective was to identify how people train for marathons, and to determine the most suitable way to categorise the various approaches athletes take when preparing for a marathon.
There were some overall trends which I discovered regarding the occurrence of training sessions, in the sense that the number of sessions peaked about a month prior to the actual marathon, and the final few days sees very little activity in terms of training as athletes begin to rest themselves for the big race. These were some of the few elements which the clusters shared.
Prior to obtaining the clusters (using feature extraction), I discussed my initial intuitive, which was that there existed three categories of athletes.
"The first group of athletes are those who don't train frequently (casually). The second group is those who train frequently but with no real pattern (hobbyists), and finally,those who run frequently and with a set pattern as to when their training session (and the extent of their training sessions) occur (elites)."
This opinion was based on the idea that an athlete's approach could be categorised by the patterns in occurrence, and the overall quantity of their training sessions. However, I discovered that this was not necessarily the case as the project developed. I found that perhaps there is a better way to categorise them, based on the features which their training programmes are based around. Here is what I have determined to be the most suitable way to categorise the various approaches taken by athletes when training for a marathon.

- The worst performing cluster of athletes in the marathon will generally be those who follow no set training plans. They are generally the worst performing cluster of athletes in all aspects of training.
    1. Number of training sessions
    2. Average weekly hours spent training
    3. Distance Covered
    4. Average speed

- The second best performing cluster of athletes in the marathon will be those athletes who follow set training programmes, regarding the distance which the cover. Athletes in this group tend to be somewhere in the middle between the other two clusters in all of the aspects mentioned above.
- The best performing cluster of athletes will be those who follow a set training plan, which is focused around both the speed at which they averaging over their sessions and the distance they cover. They are superior to their counterparts in all aspects of their training.

5.2 Project Review

Overall I am really pleased with the project and it's outcomes. It was really satisfying to see how the results I obtained complemented each other to form a bigger picture.
Regarding the process which I took to achieve these findings, it was a frustrating yet extremely rewarding journey in the end. I was disappointed initially with the plots which I was receiving from using the dynamic time warping approach. Perhaps I spent too much time trying to incorporate dynamic time warping into my project. Once assured that I had implemented it correctly and those resulting metrics were plausible, there was no more I could do. Believing that it was best to keep the level of detail in the time series, I decided to approach the task from a different angle. Feature extraction allowed me to include both athlete's distance traveled and average speed (whereas dynamic time warping only allowed for one feature to be passed e.g. distance), which meant that more information was being taken into account when plotting the data using dimension reduction techniques. We later went on to discover the importance of speed. Had I not changed my approach I would never have obtained clusters which took the average speed of athletes into account.
Having previously researched the theory behind dynamic time warping, I began to fixate about implementing it into my project. However, It just did not work the way I had hoped, and that was hard to take at the time. The thing which I will take away from this experience is that it is important to keep an open mind throughout projects, and you have to be willing to accept that not everything will pan out the way you hope it will.

Regarding the advanced section of the project, I was extremely happy with the recommender model which I have implemented. I honestly believe there is a huge potential for such an app (or a feature within Strava), especially when you compare it to existing approaches of obtaining recommended training targets. Its simple approach, the ease at which the results can be interpreted, along with the fact that its based on real proven data, make it both trustworthy and effective than currently existing approaches.

5.3 Possible Future Work

There are multiple avenues where further research could be carried out. One for example, is a similar study on a dataset with more features. I was restricted to just basing my research on primarily the distance and speed features of the athletes, as other features such as heart rate and cadence were missing for the large majority of athletes. In fact, only 47 athletes had sufficient data for all four of these mention features. Perhaps there is an advancement needed in technology (such as phones and fitbits) to accurately calculate these measurements before such a study could be carried out. I would be really intrigued to know what sort of plot this would return, and which extracted features would be the most important.
Finally, I would also like to revisit dynamic time warping as I feel that there is a potential use for this technique in the sense of accurately classifying training programmes. Unfortunately, time didn't permit for me, but it would be interesting if one could compute the dynamic time warping metrics for both multiple features (e.g. pace, distance) and perhaps weight them in an appropriate way to achieve suitable plots.

5.4 Running The Code

Permission must be sought from the project supervisor in order to gain access to the data used and the GitHub repository where the notebooks can be found. You can contact him at aonghus.lawlor@ucd.ie
Once, permission is granted the notebooks can be found HERE. Instructions on how to run the notebooks, can be found in the README file.

Bibliography

[1] Chotirat Ann Ratanamahatana and Eamonn Keogh . *Making Time-series Classification More Accurate Using Learned Constraints* . 12 pages. University of California - Riverside. Retrieved from: Click Here

[2] Meinard Mller. *Chapter: Dynamic Time Warping. In: Information Retrieval for Music and Motion.*. 15 pages.Springer, Berlin, Heidelberg. 2007. Retrieved from: Click Here

[3] Pavel Senin. *Dynamic Time Warping Algorithm Review*. 23 pages. Information and Computer Science Department, University of Hawaii at Manoa Honolulu, USA. 2008. Retrieved from: Click Here

[4] Pedregosa et al. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research. Volume 12. Pages 2825-2830. 2011. Retrieved from: Click Here

[5] Taiwo Oladipupo Ayodele. *Types of Machine Learning Algorithms, New Advances in Machine Learning*. 32 pages. ISBN: 978-953-307-034-6. Yagang Zhang (Ed.), 2010. Retrieved from: Click Here

[6] de Leeuw, J. *Multidimensional Scaling*. 12 pages. UCLA: Department of Statistics, UCLA., 2000. Retrieved from: Click Here

[7] Doherty C, Caulfield B, Davenport J, Lawlor A, Smyth B, Keogh A. *An evaluation of the training determinants of marathon performance: a meta-analysis with meta-regression*. In Press. Journal of Science and Medicine in Sport (resubmitted on request following peer-review).