Predicting Human Performance with Machine Learning

Katherine Deegan, kad2979

The University of Texas at Austin

Table of Contents

[Abstract 4](#_Toc152581164)

[Predicting Human Performance with Machine Learning 5](#_Toc152581165)

[1. Introduction 5](#_Toc152581166)

[2. Research Background 7](#_Toc152581167)

[Factors Effecting Running Performance 7](#_Toc152581168)

[Intrinsic Factors 7](#_Toc152581169)

[Extrinsic Factors 11](#_Toc152581170)

[Race Prediction Tools 11](#_Toc152581171)

[Online Race Prediction Calculators 12](#_Toc152581172)

[VDOT – an adjusted VO2 Max 14](#_Toc152581173)

[Garmin 15](#_Toc152581174)

[3. Data Pre-Processing 16](#_Toc152581175)

[Data Overview 16](#_Toc152581176)

[Feature Selection 17](#_Toc152581177)

[Handling Null Values 18](#_Toc152581178)

[Data Encoding: Handling Categorical Features 19](#_Toc152581179)

[[Heading 2] 20](#_Toc152581180)

[[Heading 3] 20](#_Toc152581181)

[Research Background 21](#_Toc152581182)

[Data 21](#_Toc152581183)

[This is the dtaat section 21](#_Toc152581184)

[References 22](#_Toc152581185)

[Footnotes 23](#_Toc152581186)

[Tables 24](#_Toc152581187)

[Figures 25](#_Toc152581188)

Abstract

[The abstract should be one paragraph of between 150 and 250 words. It is not indented. Section titles, such as the word Abstract above, are not considered headings so they don’t use bold heading format. Instead, use the Section Title style. This style automatically starts your section on a new page, so you don’t have to add page breaks. To apply any text style in this document with just a tap, on the Home tab of the ribbon, check out Styles.]

Keywords: [Tap here to add keywords.]

Predicting Human Performance with Machine Learning

# Introduction

This study explores the possibility of leveraging machine learning and personal training data to predict race performance more accurately, specifically focusing on the sport of running. Most of the tools that exist around race performance today generate a prediction either based on a previous race result or by considering another single data point, such as an individual’s VO2 Max. These tools tend to rely on vague generalizations, such as basing the prediction result on the assumption that the subject is “properly trained” [34]. Section 2 of this paper explores some of these techniques in greater detail, as well as their shortcomings.

Running is one of the most popular sports worldwide, with an estimated 50 million participants in the United States alone in 2021 [45]. The popularity of running and jogging is often traced back to the 1970s, a phenomenon often coined the “long-distance-running boom” [14] which coincided with a larger “exercise boom” that was observed across America throughout the 1960s-70s. The Boston Athletic Association Marathon, America’s most esteemed race, had just 197 participants in 1960. This number reached 1,342 by 1969 and saw nearly a sixfold increase to 7,927 runners by 1979 [32]. In 2023, the Boston Marathon had nearly 30,000 participants [2].

As the number of runners increased, so did the technology surrounding the sport. From the emergence of fitness trackers in 1965 with the Manpo-kei (the “10,000 steps meter”, invented by Japanese professor Dr. Yoshiro Hatano), to the introduction of wireless heart rate monitors in the 1980s, to the launch of modern wearables with the Fitbit in 2009, and eventually feature-rich smartwatches, beginning with the 1st generation Apple Watch in 2015 [29], personal fitness and performance data is now more accessible than ever.

With the abundance of data today, we’ve seen powerful changes in the consumer experience. Considering the healthcare industry momentarily, wearable and sensory devices have enabled 24/7 monitoring of patient metrics such as heart rate, heart rate variability, and rate of respiration, as opposed to relying on collecting this data from a single in-office visit. Patients’ health can now be analyzed more holistically, and this data can be leveraged with machine learning techniques to recognize patterns and anomalies, hopefully enabling more accurate and proactive care. This has allowed for personalization in an industry that traditionally relied on a “one-size-fits-all” approach for patient treatment [43].

Much like healthcare needs varying amongst patients, so do training needs amongst athletes [17]. The ability to better understand how your personal training effects your performance potential could be extremely valuable in a sport where world records are determined by 34 seconds [33]. \*

In the remainder of this report, we will examine popular race prediction techniques and the various factors that have been proven to correlate with running performance (section 2), the steps taken to collect training data and build a customized race prediction model (sections 3 and 4), the results of the machine learning models (section 5), and final remarks, including possibilities of future improvements to the models (section 6).

# Research Background

This section examines the intrinsic and extrinsic factors that have been proven to have a correlation with running performance, as well as some of the popular race prediction techniques that exist today and their shortcomings.

## Factors Effecting Running Performance

### Intrinsic Factors

When attempting to estimate running performance, the basis for the prediction tends to be the athlete’s intrinsic qualities. This includes both quantitative measures of fitness, as well as genetic characteristics. Three of the primary metrics used to quantify an athlete’s aerobic ability and determine their performance potential are maximal oxygen uptake (VO2 max), running economy (RE), and anaerobic threshold (AT) [52].

VO2 max is a measure of the maximum rate of oxygen delivery and utilization to your muscles during cardiovascular exercise [18]. In other words, a measure of fitness where a higher VO2 max indicates a higher degree of fitness. Many modern fitness trackers, such as Garmin watches, provide VO2 max estimations based on the relationship between the user’s heartrate and pace [18]. VO2 max can be particularly useful to pinpoint key metrics such as heart rate zones and anaerobic threshold, which can facilitate more effective training regimes [37]. Although VO2 max has a clear correlation with endurance race results, predicting an estimated 59% of the variance in marathon times [52], the absence of significant differences in VO2 max amongst top runners [26] and the observed variance in maximum oxygen uptake [29] suggests that VO2 max may not be the best indicator of one’s relative race performance.

Running Economy (RE) refers to the amount of energy required to run at a specific speed and has been shown to be a more useful performance predictor amongst a homogeneous group of runners with similar VO2 Maxes [10]. Improvements in RE will allows athletes to run at higher speeds with the same oxygen uptake [29]. RE is also highly trainable, and therefore improvements in RE is a key focus amongst coaches. Typical training interventions aimed at RE advancement include strength work, training in hot environments, high-carbohydrate diets, and training at altitude [10]. Altitude training in particular has been consistently effective in improving RE, as demonstrated by a 20-day study where 22 elite distance runners were divided into three groups, each group training at a different altitude over the course of the study (2,000-3,100m, 1,500-2,000m, and 600m, respectfully). Each of the athletes’ performance was evaluated at low altitude (600m) before and after the training interval, and it was determined the athletes training at higher altitude experienced improvements to their RE [41].

A separate study attributes the improvements in RE observed during altitude training specifically to the changes in net muscle lactate release [5]. Lactate acid is naturally produced by the body during normal metabolism and exercise, and is essential to maintaining proper bodily function, acting as the body’s primary energy source, a precursor for producing glucose, and a signaling molecule [39]. However, if the body produces too much lactic acid that it cannot remove quickly enough, severe conditions such as lactic acidosis or hyperlactatemia may be a risk. Lactate levels in the body can increase nearly twenty-fold during intense exercise, and proper training can help regulate this by allowing you to perform at a higher rate of work without raising your blood lactate levels [50]. More specifically, training results in reduced lactate production and increased lactate re-uptake within the body. Active recovery practices, or light intensity exercise following strenuous exertions, has also been shown to improve lactate removal [16].

The final metric often used to predict running performance is one’s Anaerobic Threshold, or AT. AT is a sustainable level of effort that an athlete could maintain for hours [54]. More specifically, the aerobic threshold is the point where the lactate levels start to rise in the bloodstream and anaerobic energy pathways start to assist in energy production. A higher AT means you can train at higher intensities without lactate building up [54]. Training at one’s anaerobic threshold level generally results in minimal increases to VO2 max and AT speed, but significantly increases in the time to exhaustion at AT speed, once again likely do to improving lactate level regulation [3]. In summary, a higher anaerobic threshold indicates a higher level of lactic acid tolerance, which corresponds to a higher VO2 max [10].

The intrinsic factors discussed so far (VO2 max, RE, and AT), are all trainable to some degree. The final intrinsic factors we will consider, however, are those that are unchangeable – one’s genetics.

Gender differences for one, will always have an impact on performance capabilities. Males on average have greater muscle mass, heart size, and hemoglobin concentration, meaning the VO2 max potential is higher in men than women [52], explaining the 10-12% slower race times seen by women compared to men at an elite level [21].

Age is another clear factor in performance. An athlete’s peak maximum oxygen consumption (VO2 max) is achieved around 27 years for males, and 29 years for females [24]. Peak endurance capability is generally thought to be maintained until around 35 years of age, followed by a moderated decrease until 50-60 years, and a more dramatic decrease thereafter [31]. Decline in endurance capability with age is attributed to reductions in VO2 max and lactate threshold.

Nationality also has been proven to play a role in performance ability. African runners, particularly East African, have historically had faster marathon times than non-Africans [52].

Additional genetic characteristics, such as longer legs, lower body fat, and higher flexibility, are of course going to make an individual better equip for running from the beginning [52].

### Extrinsic Factors

Likely carrying less weight on performance prediction than the intrinsic factors discussed previously are extrinsic considerations, such as climate, topography, and running equipment.

Topography, specifically elevation gain, can be a significant indicator of performance potential on a particular course. The Berlin Marathon, for example, is known to be a relatively flat course, and holds the fastest winning times by comparison [9].

Other environmental factors also appear to have a correlation with race times. The best marathon records generally occur in the autumn or springtime, with moderate temperatures ranging between 10 and 15°C [9]. Additionally, races that occur in cities with higher degrees of air pollution have shown reductions in average performance [27].

Finally, new advanced shoe technology, or NAST, became popular around 2017 when NIKE introduced their Vaporfly shoe as part of a larger project aimed at running the first sub-2-hour marathon. NAST includes shoes with carbon fiber plates, greater stake height, and lower weight. After the introduction of this technology, runners wearing NAST ran ~1% faster in the marathon compared to those who did not use it [38].

## Race Prediction Tools

Prior to building out the personalized race predication model detailed in this report, extensive research was done on the current tools and techniques that are used to predict race performance. The remainder of this section outlines how these tools work, and Section 5 compares the predictions generated by these tools with the predictions generated by the custom machine learning model.

### Online Race Prediction Calculators

Many of the online race prediction tools prompt the user to input a previous race distance and finish time and input these measures into a formula to calculate an estimated finish time for a different distance. Some of the tools also ask for additional data points, such as age or sex.

A few of the online tools offer a more detailed breakdown of how their race time prediction is derived. Marathon Handbook’s online race time calculator, for example, provides time estimations based on two formulas: the Pete Riegel and David Cameron formulas. The Pete Riegel formula was published in Runner’s World magazine back in 1977 and proposes that a linear relationship between distance and speed, where a person’s race speed declines about 6% as the distance doubles [36]. Riegel formula is one of the most popular calculation methods for many of these online tools. Equation (I) below depicts the Riegel formula, where D1 and T1 correspond to the previous race distance and time, D2 and T2 correspond to the upcoming race distance and time.

(I)

The David Cameron formula is more complex and leverages a non-linear regression model that incorporates race times of world-level athletes for distances between 400 meters to 50 miles [6]. One limitation with both of these formulas is that they were developed based on racing patterns of professional athletes, rather than the average runner. This can lead to underpredicted race times, especially for longer distances.

To compensate for the shortcomings of Riegel and Cameron formulas, the Marathon Handbook also offers a specific Marathon Race Time Calculator, as well as an Age-Grade Calculator. The Marathon Calculator aims at predicting race times for non-professional runners and is based on formulas derived by Andrew Vickers and Emily Vertosick. It also takes into consideration two previous races, as opposed to one, as well as an estimate of you weekly mileage during training. Potential problems here include the fact that weekly mileage often varies throughout the course of a training period, and there is no indication of how many weeks you have been averaging this mileage.

The Age-Grade calculator considers the runner’s age and gender and applies an “age factor” between 0 and 1 to the prediction. The calculator assumes peak running potential occurs between 20 and 30 years of age, which corresponding to an age factor of 1. The prediction is based on world record race time data and attempts to estimate the best possible time an athlete could run a certain distance based on their age. Equations (II), (III), and (IV) below are used to calculate the prediction with this tool.

(IV)

Each calculator makes certain assumptions and generalizations about a runner’s performance, and all the online race predictor tools and tend to emphasize that the predicted times assume “you have trained appropriately for said distance” [35]. Necessary training can vary amongst runners, and it is unclear what these tools consider to be proper training.

### VDOT – an adjusted VO2 Max

Where the online calculators explored above primarily focus on using past performance to predict race times, the intrinsic factors mentioned previously are also commonly used for predictions. One factor, known as VDOT, is particularly popular.

Famous running coach Jack Daniels developed VDOT as a combination of VO2 Max and Running Economy to serve as a more holistic measurement of aerobic fitness, as opposed to relying solely on VO2 Max [8]. Online tools allow you to calculate your VDOT based on a previous race result [51], and then use this value to look-up the corresponding race time predictions at various distances in a VDOT table [20]. VDOT can also be a helpful metric in determining target training pace bases (i.e. threshold pace) [7].

### Garmin

The final race prediction tool explored in this paper is a slightly more personalized tool, Garmin’s Race Predictor. Garmin watches have a built-in race prediction feature that primarily uses your VO2 Max, as well as training history from the past several weeks, to give you an estimate of predicted paces for several different race distances [12]. The VO2 Max that Garmin watches use for this estimation are calculated based on the relationship between the user’s heartrate and pace over recent activities [18]. This VO2 Max estimation does not consider other factors that may affect the relationship between heart rate and pace, such as illness, dehydration, lack of sleep, travelling, or running in heat or at altitude.

Limitations with Garmin’s method of race prediction include that fact that, as stated earlier, VO2 Max alone is not always the most accurate race predictor across individuals. Additionally, the VO2 Max calculation for Garmin watches rely on the accuracy of one’s heart rate measurement. Additionally, studies have found wrist-based heart rate monitors to have a margin of error ranging from 1 to 13.5% [49], which often leads to underestimations of heart rate, and thus overestimations of VO2 Max and race pace. Also, as with the other race prediction methods mentioned, Garmin’s calculation assumes that you have completed proper training for that race distance [13] and does not consider any additional factors such as weather, course difficulty, terrain, or training regime.

# Data Pre-Processing

Common limitations between all the race prediction methods explored in Section 2 are that the prediction is not personalized to the individual, nor does it consider any extrinsic factors. The individual’s specific training patterns (mileage; running pace - intervals, threshold, easy miles, etc.; training at elevation; training at heat; cross-training), level of effort they may put into a race activity vs their potential (do they run to exhaustion during a race, or do they maintain a comfortable level of exertion), anomalies (such as injuries, illnesses, amount of sleep) all may effect an athlete’s performance differently, and to varying degrees. Additionally, the extrinsic factors unique to the environment where the race is occurring (temperature, altitude, elevation) will all impact the level of difficulty of the course, and thus the individual’s performance. The data used to train the personalized, machine learning-based model explored in this project was specifically selected and curated to compensate for these shortcomings.

## Data Overview

Strava is a comprehensive, sport-centric social media platform with features that enable athletes to track their own training progress, connect with other athletes, explore new places, and compete for leaderboard positions and in virtual competitions [46]. Most of Strava’s features are enabled by its activity tracking capabilities. Strava allows you to track a range of athletic activities either from the app itself, or from a third-party fitness tracker or smartwatch (i.e. Garmin watch, Apple watch) and upload them to your Strava profile, memorializing this activity as part of your athletic journey. Once the activity is uploaded to your profile, you can holistically review this activity by analyzing key metrics including speed/pace, elevation, distance, heart rate, and cadence, as well as share this activity with your followers [46].

Strava offers a publicly available API that allows developers to access Strava data [47]. The data set we will be using is my personal Strava activity data over the last approximately 2 years. Additionally, we enriched the activity dataset to include weather data for each specific location/activity by leveraging Meteostat’s Python library, which provides access to open weather and climate data via Pandas DataFrame objects [28]. By using one’s personal data to build out a race predication model, we can consider most of the intrinsic and extrinsic factors effecting performance that were introduced in Section 2, and hopefully build a more accurate, individualized race predication tool.

Prior to any of the data preparation steps detailed in this section, Strava’s activities API was called to retrieve my recent activity data starting from mid-2021 until the present. data was then loaded into a DataFrame using Python’s Pandas library. The code used to retrieve and prepare the data, as well as build out the models discussed in Section 4, can be found in the GitHub repository referenced in the footnotes.

## Feature Selection

Strava’s activities API returns 55 attributes corresponding to a given activity [47]. Initial feature selection was determined by the presence of null values in each column, removing one of two closely coupled columns (i.e. average pace and moving time), and focusing on the key determinants of run performance that were introduced in Section 2. Selected features included local date-time, distance, moving time, total elevation gain, activity type (i.e. Run, Bike, Swim, etc.), starting latitude/longitude, maximum speed, average cadence, average heart rate, maximum heart rate, and elevation high.

After initial feature selection, the local start date-time and the coordinates of the activity were leveraged with the Meteostat Python library to retrieve weather station data for each activity and enrich the data with an average temperature feature. The decision to include this feature was based on the proven correlation between temperature, heart rate, and running pace [52].

Another key step of our data enrichment involved adding an additional column “race”. This column would hold Boolean values indicating whether an activity was a race activity (True) or not (False). The dataset had 5 race activities, and thus 5 True values in the “race” column.

## Handling Null Values

Handling missing values is a key step in data pre-processing, as most machine learning models and Python libraries are unable to handle these Null or NaN values on their own. Missing values can also lead to biases, and thus inaccuracies, in the final model [44]. The null values in our data set were limited, but there were still a few attributes that needed to be addressed. Below is a summary of the initial null values by column in our data set.

**Table 1**

*Null Values by Feature Column*

|  |  |
| --- | --- |
| **Column Name** | **Number of Null Values** |
| start\_date\_local | 0 |
| start\_latlng | 7 |
| distance | 0 |
| moving\_time | 0 |
| total\_elevation\_gain | 0 |
| type | 0 |
| max\_speed | 0 |
| average\_cadence | 16 |
| average\_heartrate | 15 |
| max\_heartrate | 15 |
| elev\_high | 7 |
| avg\_temp\*\* | 20 |

*Note.* The table above details the number of null values present for each selected feature after the data was pulled from Strava’s API.

Many strategies exist for handling null values, but due to the ordered nature of the dataset (activities are sequenced by timestamp), it was determined that interpolation would be a sufficient method. Interpolation estimates the missing values based on the values of surrounding data points [40], and thus we are assuming that nearby activities occurred at a similar location and at a similar level of effort. The built-in interpolate method in the Pandas library was used to apply interpolation to each column with missing values [30].

## Data Encoding: Handling Categorical Features

Categorical values are those that are discrete and non-continuous. There are both ordinal (ordered) and nominal (unordered) categorical values, each of which need to be preprocessed differently [23]. In the case of our data set, the only categorical feature that would ultimately be included in our final model was the timestamp column: start\_date\_local.

Date and time were important factors to consider in our model because one’s training and performance my vary based on time of day and throughout the year. Date and time are unique categorical fields, as they are cyclical in nature. To preserve the cyclicality, Eryk Lewinson’s method of feature transformation via sine and cosine functions was used to encode the activity’s time of day, day of the month, and month of the year [25]. The following figures show the resultant sine/cosine values of each feature respectfully following the transformation.

**Figure 1**

*Time of Day Encoded*

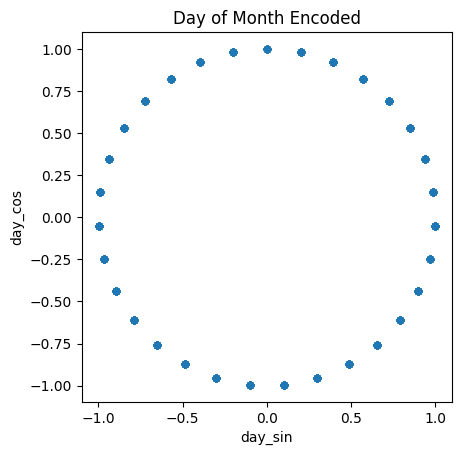
A graph with blue dots

Description automatically generated

*Note*. Seconds of the day encoded cyclically and plotted with Python’s Matplotlib library. Own work.

**Figure 2**

*Day of Month Encoded*



*Note*. Day of the month encoded cyclically and plotted with Python’s Matplotlib library. Own work.

**Figure 3**

*Month of Year Encoded*

A graph of a number of years

Description automatically generated with medium confidence

*Note*. Month of the year encoded cyclically and plotted with Python’s Matplotlib library. Own work.

Lastly, we handled year via one-hot encoding, as there were only three different year values in our dataset. Columns “2021”, “2022” and “2023” were added to the dataset, and their value was set as True or False accordingly.

## Data Encoding: Handling Numerical Features

Most of the attributes in our dataset are numeric. However, each column’s numerical values are at a different scale, meaning each column needs to be standardized to prevent columns with bigger values from dominating decisions made by our machine learning models.

There are two common methods of transforming numerical values – standardization and normalization. Normalization involves scaling the column values on a set range, typically between 0 and 1 or -1 and 1. Normalization is useful when the distribution of a column is unknown or not normal, but normalized columns tend to be more effected by the presence of outliers [19]. Equation (V) below demonstrates how some of the columns in this study were normalized on a scale of 0 to 1, where *X* represents the raw score, and *min(x)* and max(x) represent the minimum and maximum values of the column, respectfully.

Standardization, on the other hand, involves scaling a numeric column so that the data has a mean of 0 and standard deviation of 1. Standardization is typically the preferred method for normally distributed values and can be accomplished by calculating the z-score of each value using the equation (VI) below. Here, *X* represents the raw score, represents the mean value of the column, and represents the column’s standard deviation.

In the case of our data set, we first looked at the distribution of values across each numeric column and then made the decision on whether to use standardization or normalization to transform the values. The distribution of values across each column are illustrated in the figures below.

**Figure 4**

*Average Heartrate Distribution of Values*

A graph of a number of heartrate distribution

Description automatically generated

*Note*. The Average Heartrate values and their frequency was plotted with Python’s Matplotlib library. The mean value was 149.26, and the standard deviation was 9.10. Own work.

**Figure 5**

*Maximum Heartrate Distribution of Values*

A graph of a number of heart rate

Description automatically generated

*Note*. The Maximum Heartrate values and their frequency was plotted with Python’s Matplotlib library. The mean value was 171.22, and the standard deviation was 10.50. Own work.

**Figure 6**

*Maximum Speed Distribution of Values*

A graph of a speed distribution

Description automatically generated

*Note*. The Maximum Speed values and their frequency was plotted with Python’s Matplotlib library. The mean value was 5.36 and the standard deviation was 1.71. Own work.

**Figure 7**

*Elevation High Distribution of Values*

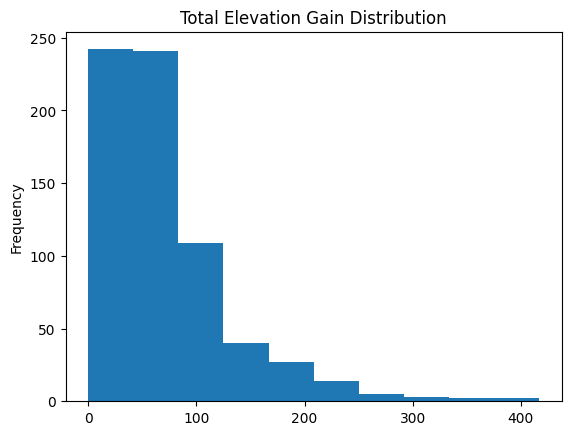
A graph of elevation high distribution

Description automatically generated

*Note*. The Elevation High values and their frequency was plotted with Python’s Matplotlib library. The mean value was 290.65 and the standard deviation was 417.36. Own work.

**Figure 8**

*Total Elevation Gain Distribution of Values*



*Note*. The Total Elevation Gain values and their frequency was plotted with Python’s Matplotlib library. The mean value was 71.34 and the standard deviation was 59.91. Own work.

**Figure 9**

*Average Cadence Distribution of Values*

A graph of a number of blue bars

Description automatically generated with medium confidence

*Note*. The Average Cadence values and their frequency was plotted with Python’s Matplotlib library. The mean value was 49.41 and the standard deviation was 1.80. Own work.

**Figure 10**

*Average Temperature Distribution of Values*

A graph of a temperature distribution

Description automatically generated

*Note*. The Average Temperature values and their frequency was plotted with Python’s Matplotlib library. The mean value was 20.23 and the standard deviation was 8.54. Own work.

Standard deviation is an indication of how close the values of a dataset are to the mean value. A lower standard deviation indicates a smaller distribution of values, i.e. values are closer to the mean. A normal “bell curve” distribution is recognized by a symmetrical distribution of values, where the mean is 0 and standard deviation is 1. Columns average\_cadence, max\_speed, max\_heartrate, average\_heartrate, and avg\_temp exhibited relatively symmetric, normal distributions with lower standard deviations, and thus were standardized via z-score. Columns total\_elevation\_gain and elev\_high, had more of a skewed distribution, and were thus transformed via normalization.

# Method

After pre-processing, the encoded data was used to build a series of machine learning models, the results of which are summarized and compared in Section 5. This section will walk through the model selection process and how the accuracy of the models will be assessed.

## System Overview

Two machine learning models were constructed based on the data set described above – a Linear Regression model and a Neural Network. These models were both trained on the same data; a subset of my personal Strava activity from the last approximately 2 years cleansed and encoded following the methods described in Section 3. The output of each model was a prediction of the total moving time of a running activity. The features used to train the models are summarized in the Table 2 below.

**Table 2**

*Model Training Features UNITs*

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Encoded** |
| distance | Total distance (meters) of activity | N |
| total\_elevation\_gain | Total elevation gain over the course of activity | Y |
| max\_speed | Maximum speed run during activity | Y |
| average\_cadence | Average cadence during activity | Y |
| average\_heartrate | Average heart rate during activity | Y |
| max\_heartrate | Maximum heart rate during activity | Y |
| elev\_high | The highest elevation recorded during activity | Y |
| avg\_temp | Average temperature of the location/day of activity | Y |
| race | Boolean value indicating if activity is a race activity (T) or not (F) | N |
| month\_sin, month\_cos, day\_sin, day\_cos | Cyclically-encoded columns to represent month and day of activity | Y |
| sec\_sin, sec\_cos | Cyclically-encoded columns to represent time of day of activity | Y |
| 2021, 2022, 2023 | One-hot encoded columns indicating the year of the activity | Y |

*Note.* The table above details the dataset used to train the prediction models after pre-processing.

To test each of these models, I used two of the most recent races I ran as inputs and compared the models’ predictions to my actual results. The first race, the Richmond Marathon, occurred on November 11th, 2023 in Richmond, Virginia. The second race was a 15K in Seattle, Washington on November 23rd, 2023. For each race, input features regarding distance, elevation, and date/time were all known. Features regarding speed, cadence, and heartrate were populated in the code using the average values of those respective fields from other race activities in the training data set (i.e. where “race” == True).

A special case was used to handle the average temperature. The average temperature value corresponds to the average temperature at a given location on that day the previous year. This logic was built into the code with the assumption that this race predication model would be used the predict a race result in the *future*, and thus the exact temperature on the day of the race was unknown.

For each race result prediction, the training data used to build the model spans from August 2021 to the day before the race being predicted.

## Linear Regression Modelling

The first series of models constructed were multiple Linear Regression models. Multiple linear regression is a very popular supervised machine learning technique where multiple independent variables (the input features described above) are used to predict the outcome of a single dependent variable (total moving time) by establishing a linear relationship between the two. A linear regression algorithm finds a best-fit line between the independent and dependent variables that minimizes the sum of squared errors between the actual and predicted values [4]. Equation (VII) shows the regression line equation where *Y* is the dependent variable, *x1, x2, … xn* are the independent variables, and , , … are coefficients.

VII)

## Neural Network Modelling

The second series of models were Neural Networks. This is a type of deep learning that is loosely modelled after the human brain and consists of upwards of thousands densely interconnected nodes [Hardesty]. Typically, these nodes are organized into layers, and the layers pass data between them. Oftentimes a single node is receiving information from multiple nodes in the layer proceeding it and is also sending data to multiple nodes in the layer following it.

The neural network models were built using Python’s Keras library. This is part of the larger TensorFlow Python library, and allows you to develop and build deep learning models with minimal code [keras].

# Results

This section compares the results and accuracy of the machine learning models created during this study.

## Actual Race Results

Table 3 summarizes my actual race results from the marathon and 15K. These times will be compared to the models’ predicted race time to assess accuracy.

**Table 3**

*Actual Race Results*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Race Name** | **Race Date** | **Distance**  (miles) | **Total Moving Time**  (HH:MM:SS - as recorded by Garmin) | **Pace**  (min/mile) |
| Richmond Marathon | 11/11/2023 | 26.2 | 3:05:01 | 7:03 |
| Seattle Turkey Trot 15K | 11/23/2023 | 9.32 | 1:03:59 | 6:52 |

*Note.* The table above details the actual recorded time for each race by my Garmin watch.

## Prediction from Existing Tools

Prior to evaluating the race predictions generated by the custom machine learning models, race estimations for the marathon and 15K were generated using the existing tools described in Section 2. The prediction results from these tools, as well as my past race results that were supplied to the tools when applicable, are summarized in the tables below.

**Table 4**

*Previous Race Results*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Input Race** | **Date** | **Distance** (miles) | **Chip Time**  (HH:MM:SS) | **Pace** (min/mile) |
| **1** | 3M Half Marathon | 1/22/2023 | 13.1 | 1:26:28 | 6:36 |
| **2** | Austin Half Marathon | 2/19/2023 | 13.1 | 1:29:42 | 6:51 |
| **3** | Dallas Turkey Trot 5K | 11/24/2022 | 3.1 | 00:19:17 | 6:13 |

*Note.* The table above show the previous race results that were used as inputs to the existing online race prediction tools, where applicable.

**Table 5**

*R*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prediction Tool** | **Input Race(s) Used** | **Extra Informarion** | **Predicted Marathon Time**  (HH:MM:SS) | **Predicted Marathon Pace**  (min/mile) | **Predicted 15K Time**  (HH:MM:SS) | **Predicted 15K Pace**  (min/mile) |
| [SportTracks](https://sporttracks.mobi/labs/race-finish-time-predictor) | 1 | Sex (F) | 03:07:05 | 7:08 | 1:00:14 | 6:28 |
| [Marathon Handbook](https://marathonhandbook.com/race-time-calculator/)  (Pete Riegel) | 1 |  | 03:00:24 |  | N/A | N/A |
| [Marathon Handbook](https://marathonhandbook.com/race-time-calculator/)  (David Cameron)  [Marathon Handbook](https://marathonhandbook.com/race-time-calculator/)  (Marathon Race Time Calculator) | 1 |  | 03:03:53 |  | N/A | N/A |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

*ace Prediction Results*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Prediction Tool** | **Input Race(s)** | **Extra Informaion** | **Predicted Marathon Time** | **Predicted Marathon Pace** | **Predicted 15K Time** | **Predicted 15K Pace** |
| <https://sporttracks.mobi/labs/race-finish-time-predictor> | **1** | **Sex (F)** | **3:07:05** | **7:08** | **1:00:14** | **6:28** |
| <https://marathonhandbook.com/race-time-calculator/>  **(Pete Riegel formula)** | **1** |  | **3:00:24** |  | **N/A** |  |
| <https://marathonhandbook.com/race-time-calculator/>  **(David Cameron formula)** | **1** |  | **3:03:53** |  | **N/A** |  |
| <https://marathonhandbook.com/marathon-race-time-predictor/>  **Marathon Race Time Prediction** | **1,3** | **Weekly Mileage: 55 miles** | **3:08:39** |  | **N/A** |  |
| <https://marathonhandbook.com/age-grade-calculator/>  **Age-Grade Calculator** | **1** | **Age: 24**  **Gender: F** | **2:58:27** |  | **1:00:41** |  |
| <https://vdoto2.com/calculator>  **VDOT** | **1** | **VDOT: 53.4** | **3:00:28** | **6:53** | **1:00:05** | **6:27** |
| Garmin Race Predictor | **N/A** | **Personal VO2 Max: 64 (as measured by Garmin watch)** |  |  |  |  |

*Note.* The table above shows the predicted race results for each of the distances in question (Marathon and 15K) using existing tools introduced in Section 2.

Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

Each calculator makes certain assumptions and generalizations about a runner’s performance, and all the online race predictor tools and tend to emphasize that the predicted times assume “you have trained appropriately for said distance” (“Race Time Calculator”, n.d.). Necessary training can vary amongst runners, and it is unclear what these tools consider to be “proper training”.

## [Heading 2]1

[To update the table of contents (TOC), apply the appropriate heading style to just the heading text at the start of a paragraph and it will show up in your TOC. To do this, select the text for your heading. Then, apply the style you need.]

[Heading 3]. [Include a period at the end of a run-in heading. Note that you can include consecutive paragraphs with their own headings, where appropriate.]

[Heading 4]. [When using headings, don’t skip levels. If you need a heading 3, 4, or 5 with no text following it before the next heading, just add a period at the end of the heading and then start a new paragraph for the subheading and its text.] (Last Name, Year)

[Heading 5]. [Like all sections of your paper, references start on their own page, as shown on the page that follows. The body of the References section uses the Bibliography style. For more detailed information on formatting references, see the APA Style Manual, 6th Edition.

# Research Background

Data

This is the dtaat section

References

Last Name, F. M. (Year). Article Title. *Journal Title*, Pages From - To.

Last Name, F. M. (Year). *Book Title.* City Name: Publisher Name.

Footnotes

1[Add footnotes, if any, on their own page following references. For APA formatting requirements, it’s easy to just type your own footnote references and notes. To format a footnote reference, select the number and then apply the Footnote Reference. The body of a footnote, such as this example, uses the Normal text style. (Note: If you delete this sample footnote, don’t forget to delete its in-text reference as well.)]

Tables

Table 1

[Table Title]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Head | Column Head | Column Head | Column Head | Column Head |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: [Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style to get the spacing between table and note. Tables in APA format can use single or 1.5 line spacing. Include a heading for every row and column, even if the content seems obvious. To insert a table, on the Insert tab, tap Table. New tables that you create in this document use APA format by default.]

Figures



Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.