Predicting Run Performance

Using fitness watches and statistical modeling to maximize performance

Joe Martin

2021-12-12

Through the past several decades, analytics have become increasingly important across all sports and at all levels, from high school recruitment, to professional play, to amateurs using devices to track fitness. With greater accessibility to smart devices that accurately record biometric and GPS data, all athletes have the opportunity to utilize analytics to enhance their performance. This paper investigates how individuals can utilize data from smart devices to predict average pace while training for road races of any distance, including marathons and half-marathons. Having the ability to accurately predict average pace gives runners the opportunity to tune their performance and tailor their training plans.

# 1 Introduction

## 1.1 General Background Information

Marathon and distance running requires a special amount of detail and planning for athletes to achieve their goals - whether it’s completing their first race or achieving a new personal best. With smart watches becoming more accessible to athletes of all abilities, everyone now has access to a constant stream of data they couldn’t imagine having in the past, including an accurate average pace, heart rate, and even measurements like stride length for individual activities.

The goal of this project is to create a model which can take in variables from recent activities and predict a potential race time - as in, if I were to run a marathon today, under certain conditions, how would I perform? The goal of this project is to create a proof-of-concept model which, when deployed, can take in variables from the user and predict their average pace.

As an additional benchmark, it is ideal to limit error as much as possible. This is important because average pace can be easily affected by small changes in time. For example, if an athlete runs a marathon at a 7:00 pace, their final race time is 3:03:32. If an athlete runs a race at a 7:05 pace, their final race time is 3:05:43. While this doesn’t seem like much of a difference, this could easily be the difference between comfortably earning a good qualifying time for a future race and missing the qualifying time for the same race.

Therefore, the goal for this project is to limit error to be within 5 seconds of the true average pace.

## 1.2 Description of data and data source

The data used for this model comes from my Garmin Forerunner 245 and run journal. The final data set contains 417 observations and more than 20 variables. Variables include date, distance, weather (temperature and conditions), total ascent, total descent, min elevation, max elevation, cadence, average stride length, and average pace. Much of this data is replicated in my running journal. While my running journal isn’t included in the model, it was used in my data exploration and pieces of it were joined with my final model set.

## 1.3 Questions/Hypotheses to be addressed

Investigate whether it is possible to predict average pace with low error (within 5 seconds).

# 2 Methods

## 2.1 Data aquisition

All data used in this project comes from Garmin, but was obtained in two ways. Garmin offers users an online dashboard called Garmin Connect, which houses all data recorded from Garmin devices. Most of the data used in this project comes from the .csv export feature in the dashboard. However, there were several variables of interest included in the dashboard which weren’t available for download, including average speed (mph), max speed, and anaerobic training effect measurements (a measurement created by Garmin to estimate how much an activity would impact anaerobic fitness). For these variables, I built a web scraper using the RSelenium package and joined the resulting dataset to the primary Garmin data.

In addition to all Garmin data, some variables from my running journal were included. These include temperature and weather conditions. As noted previously, much of this data was processed in anticipation of using it for this project. However, there were too few features to include this data in statistical models, so while it appears in my processing script and exploratory analysis, it was not utilized further.

## 2.2 Data import and cleaning

In order to import and clean my data, I simply used read\_csvfor my main Garmin file. In order to import my training journal data, I used the googlesheets4 package to add data to my processing script, then saved these imports as .Rds files in order to keep my Google account secure. Much of this data was already relatively clean and in tidy format. Major changes to this data involved removing variables I knew I would not use, like variables captured by my watch during swims or bike rides. Otherwise, I took care to ensure all data were coerced to the proper type.

### 2.2.1 Data Preparation

Perhaps the biggest consideration I needed to make during data preparation was how to format my average pace variable, as well as other time variables. After much testing and consideration, I ultimately converted my average pace to be a double with the precise number of seconds. For example, instead of showing a 6:30/mile average pace, my dataset will show an average pace of 390 seconds per mile. Too many errors were occuring when formatting this values with lubridate and base functions like as.POSIXct() for this value to be useful in modeling if it were in mm:ss format. Since average pace is the target variable for modeling, it was important to have this in a consistent, accessible format.

### 2.2.2 Exploratory Analysis

My exploratory analysis began with analyzing ordinary least squares (OLS) using the lm() function to see if any variables were correlated. I graphed these relationships, as well, in order to better visualize this relationship. With average pace as my target variable, I focused on plotting variables in relation to it. As expected, average heart rate, cadence, and aerobic training effect had strong relationships with average pace. Average Speed also had a particularly strong relationship with average pace. However, this variable was eliminated due to its potential to cause data leakage.

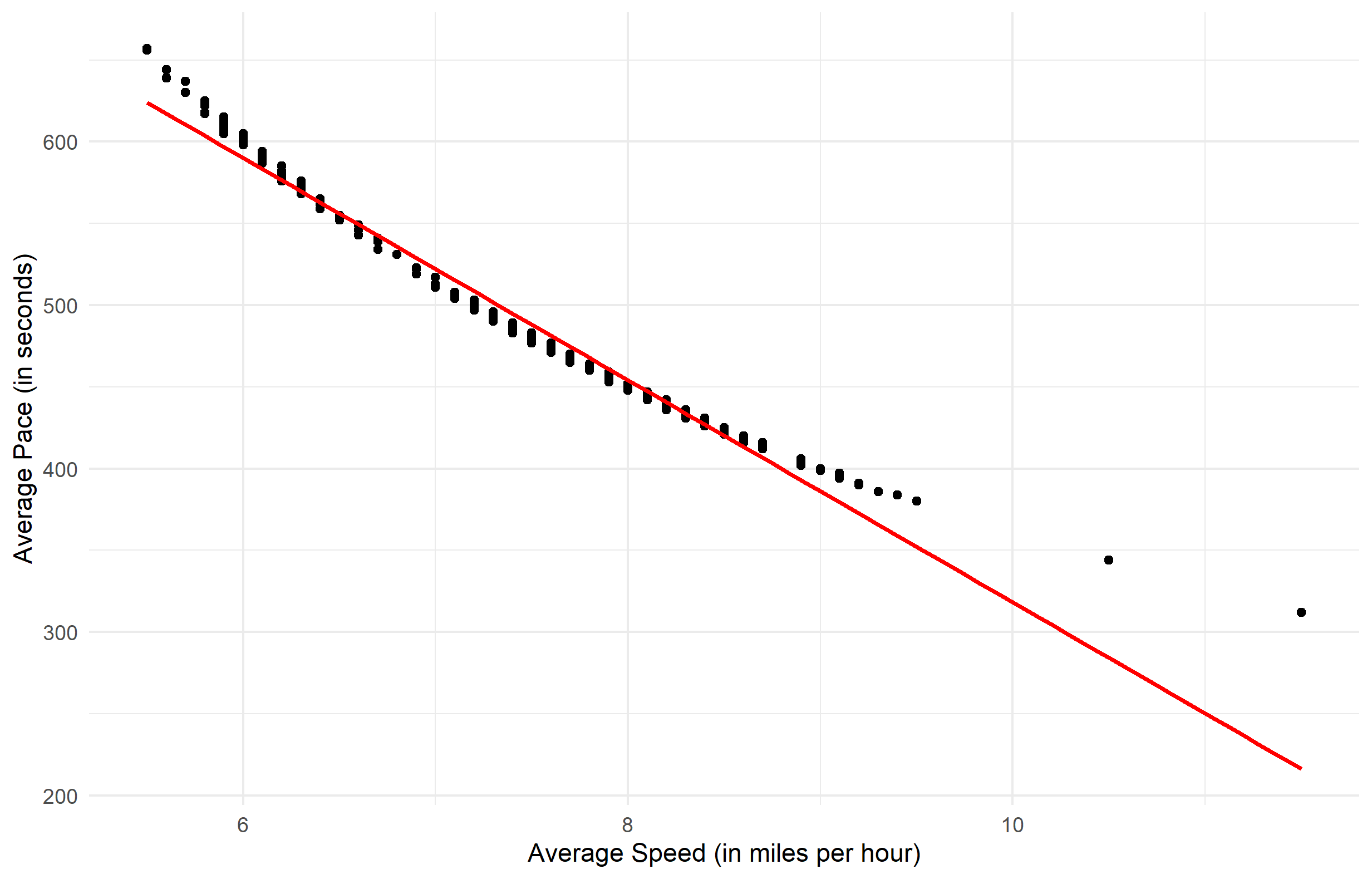


Figure 2.1: Average Speed is clearly related strongly with average pace. However, since this variable is so closely related and serves almost the same purpose as average pace in measuring running speed, it should not be used for modeling.

#### 2.2.2.1 Average Pace

##### 2.2.2.1.1 Aerobic Training Effect

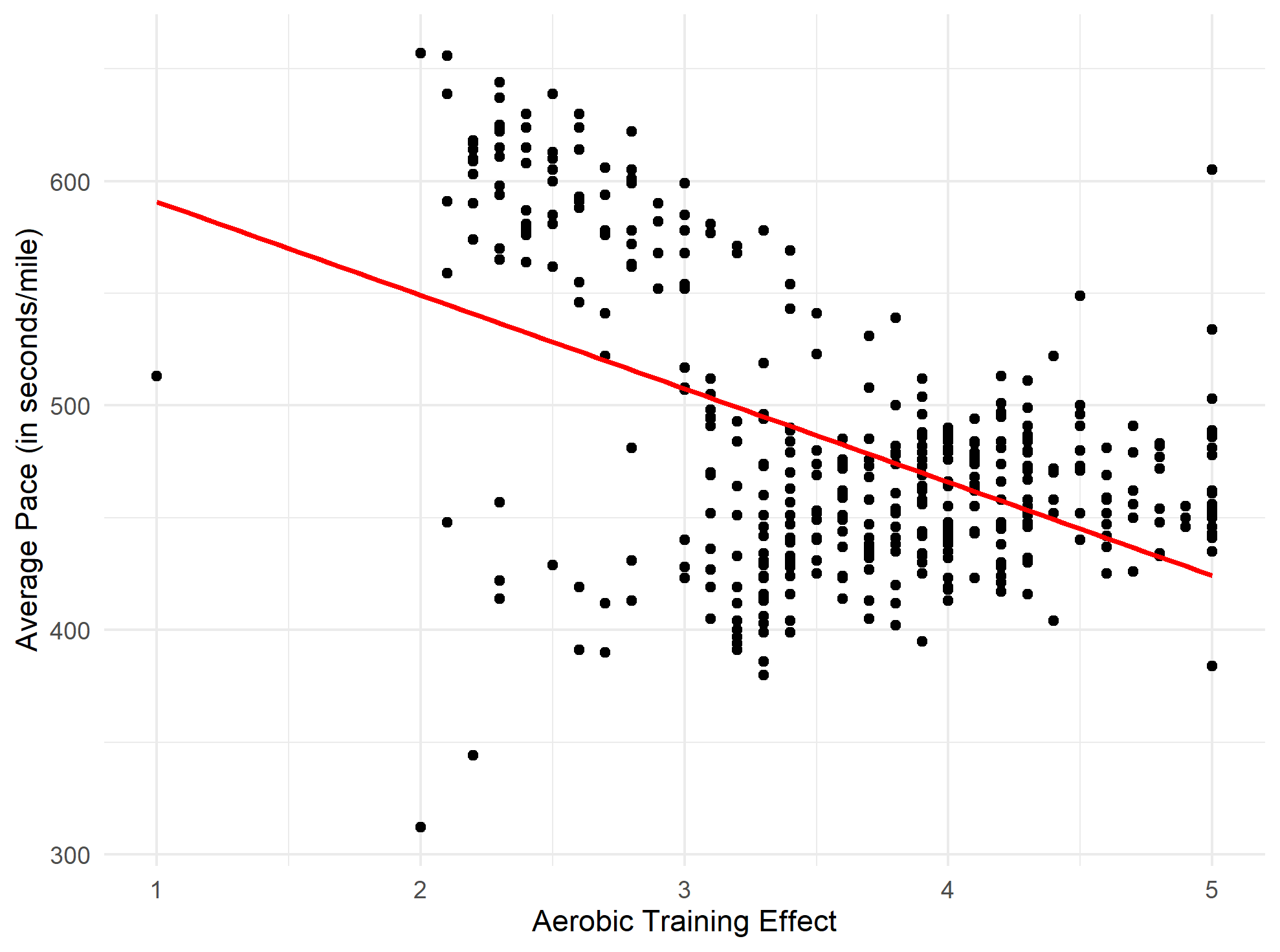


Figure 2.2: Aerobic Training Effect (Aerobic TE) is a measurement Garmin created to rate an aerobic effort. On a scale from 1-5, 5 is a maximum effort. 5 is typically associated with hard efforts like races and longer threshold runs. 1 is minimal and typically associated with other types of exercise besides running.

##### 2.2.2.1.2 Average Heart Rate

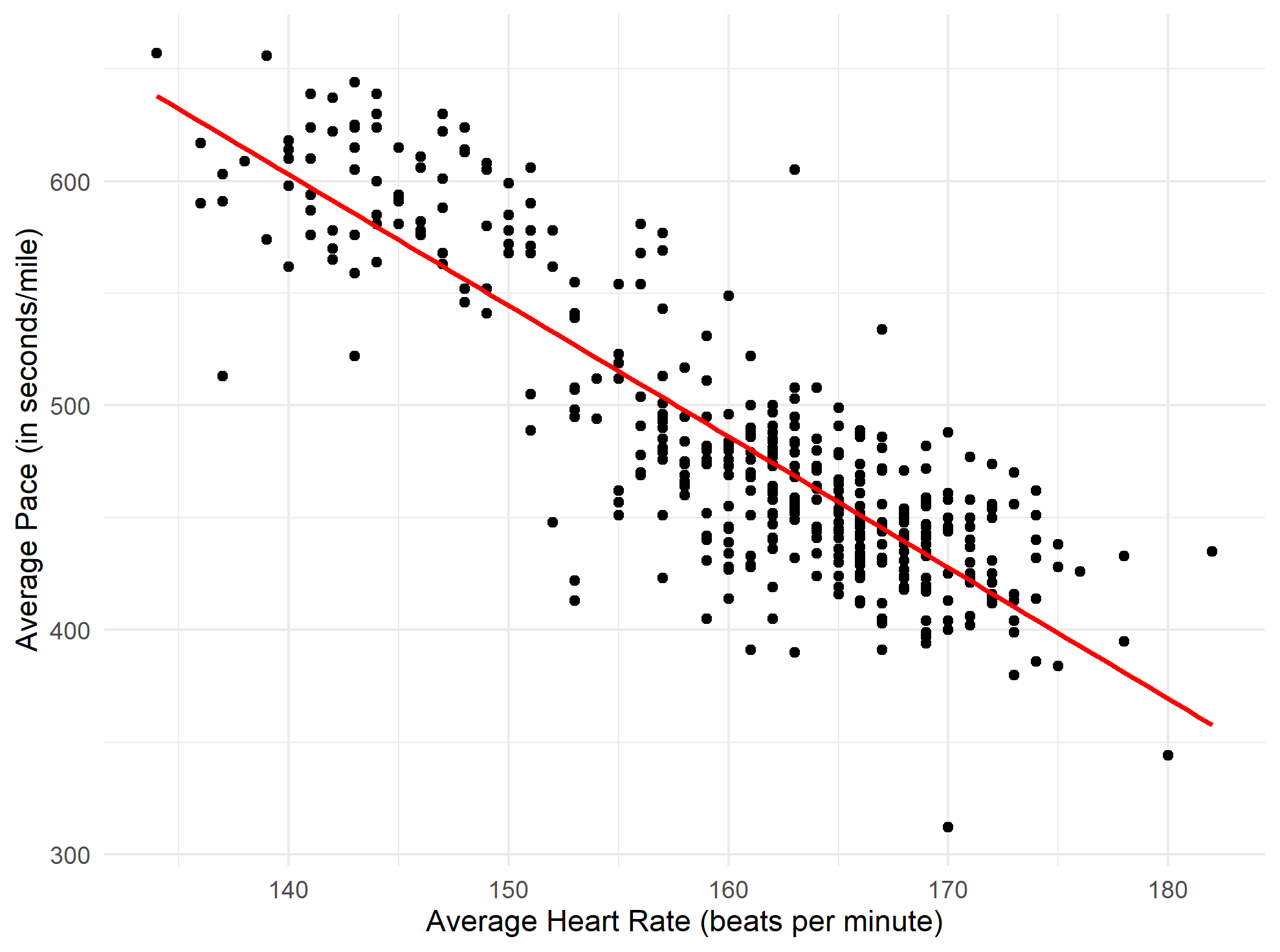


Figure 2.3: Average heart rate is perhaps the most obvious variable to use as a predictor. It is also a useful variable for athletes since heart rate data is accesible during workouts through smart watches.

##### 2.2.2.1.3 Stride Length

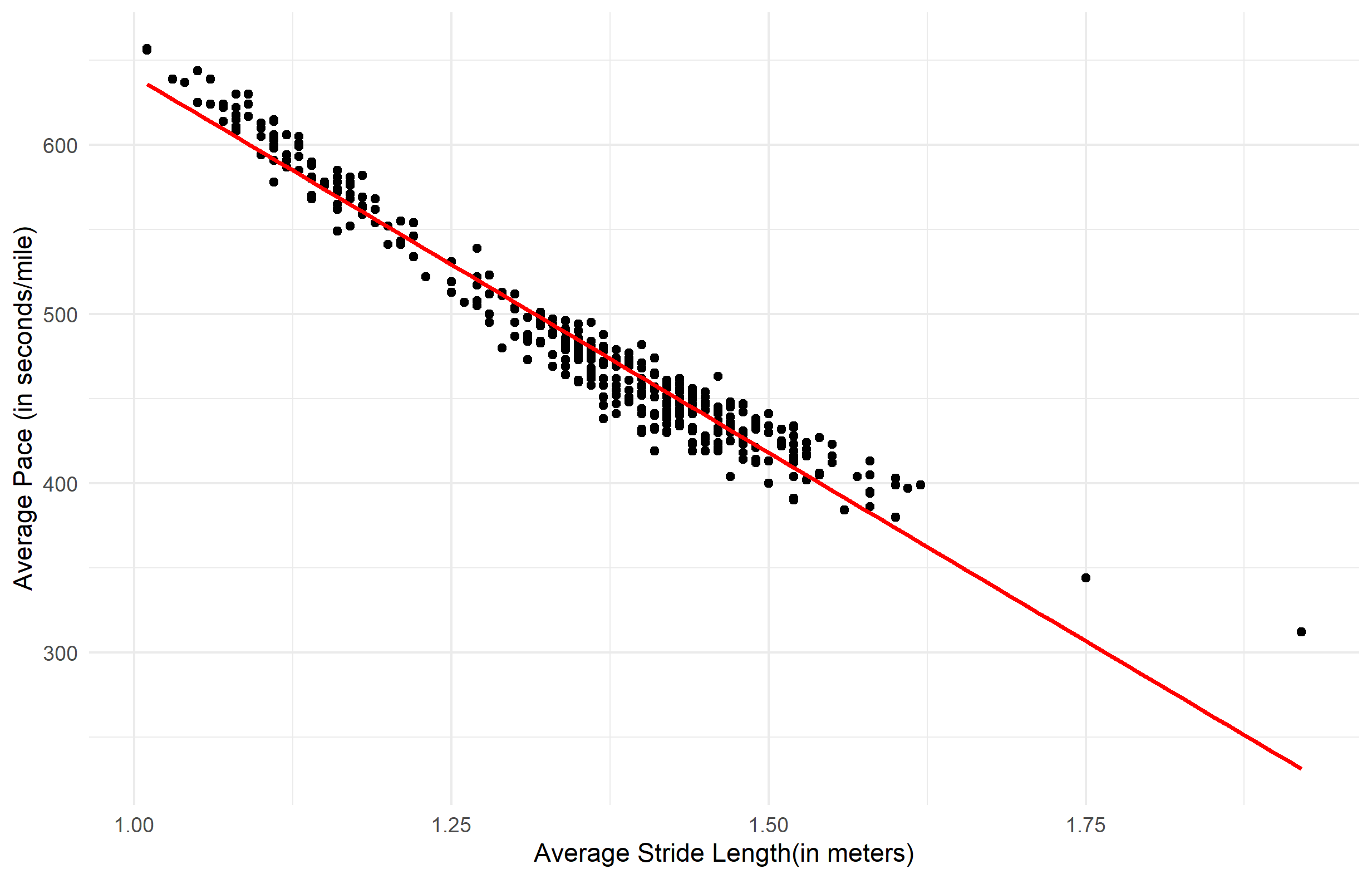


Figure 2.4: Greater stride length seems to be associated with a greater pace. This seems counter-intuitive if thinking about cadence in this equations, but with good form and mechanics, greater cadence is more likely to be associated with greater stride length.

##### 2.2.2.1.4 Cadence

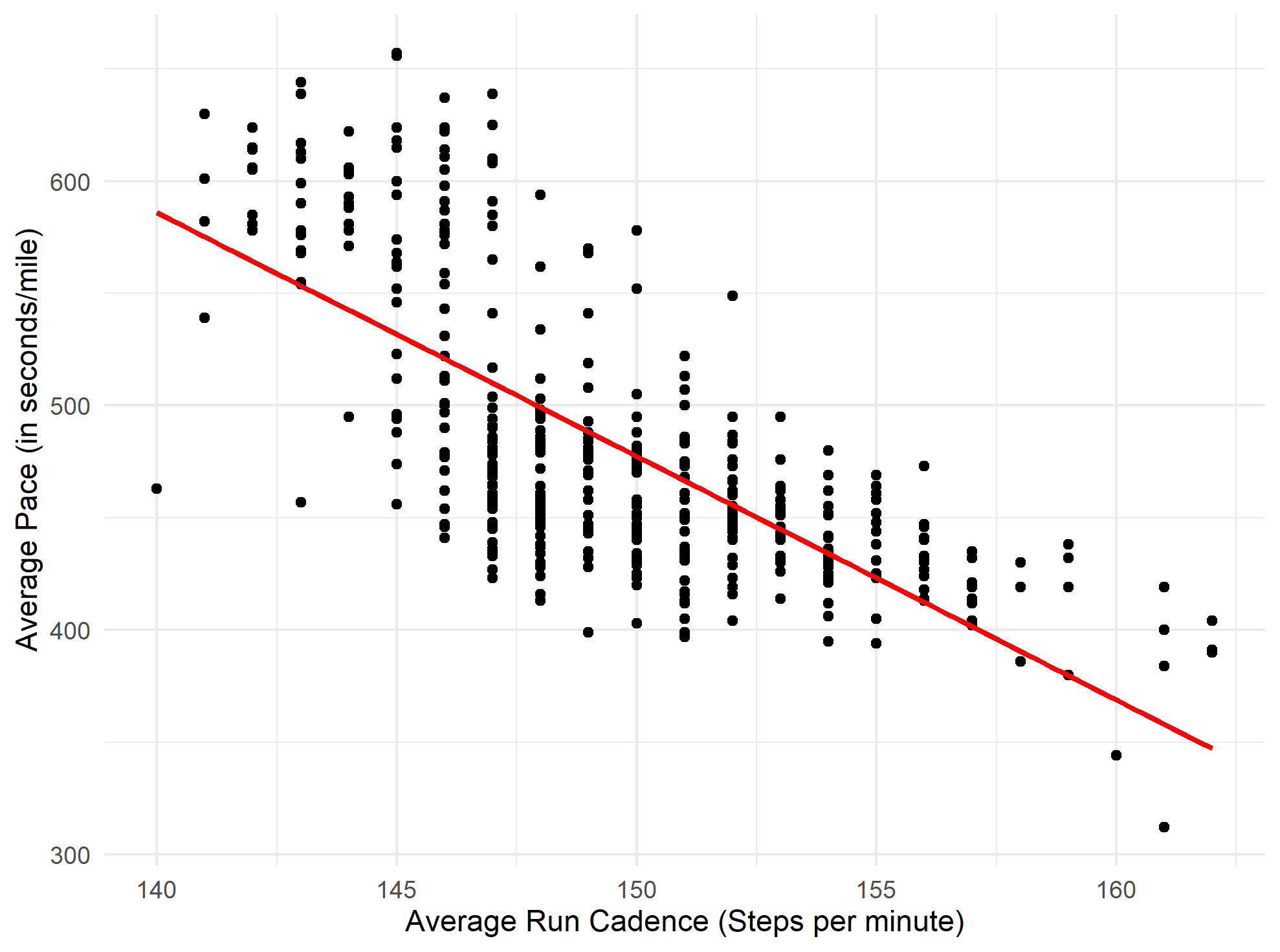


Figure 2.5: Cadence is the measure of how many times a runner’s feet hit the ground per minute. A high cadence is typically associated with a faster pace.

## 2.3 Statistical analysis

### 2.3.1 Initial Process

The goal of this project is to investigate whether data from a smart watch can be used to predict average pace. More specifically, the objective is to predict average pace within 5 seconds. Therefore, the target variable is average pace. In order to begin statistical modeling, OLS regression was used to review strong relationships between average pace and all predictor variables. This yielded the same results as the exploratory analysis. Average heart rate, average cadence, average stride, and Aerobic TE had the strongest relationships, as demonstrated by their p-values.

The tidymodels framework was used in order to conduct all statistical analyses. While the first model I produced was a simple linear regression with an R Mean-Squared Error (RMSE) value less than 5. However, when I tried improving this performance with a random forest, I realized I had used the average speed variable in my analysis. Using the vi function from the vip package, I was able to determine the average speed variable had the highest importance and that it likely had too much influence on my model.

After this adjustment, I began with a simple linear regression with the lm engine. Initial results yielded an estimated R Mean-Squared Error of 9.6, which fails the target of 5. However, this seemed to be a good starting point for building a random forest and to use v-fold cross-validation to improve results.

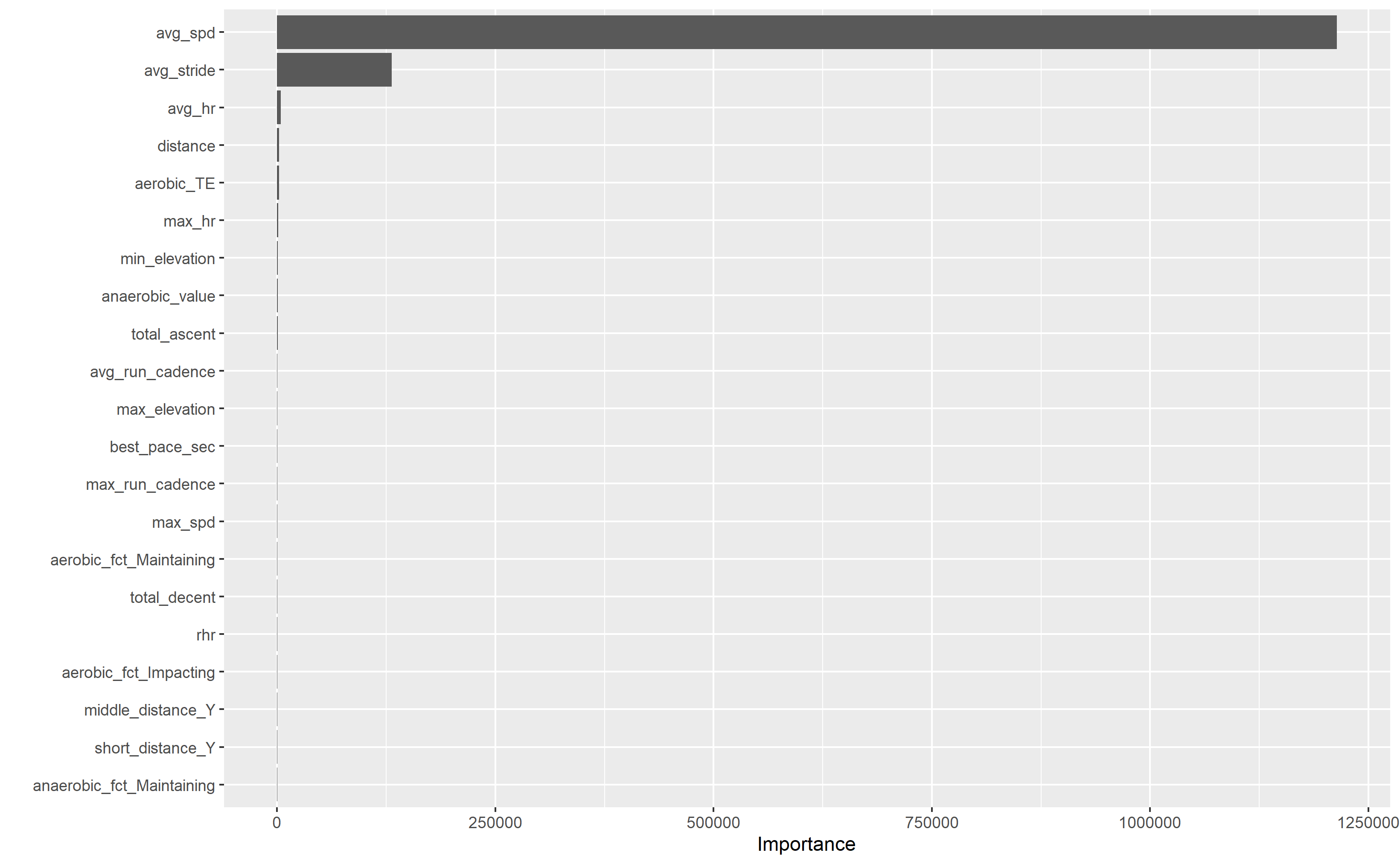


Figure 2.6: This figure shows the importance each variable has to the resulting model. Average Speed seems to cause data leakage, resulting in the model providing unrealistically good predictions.

# 3 Results

## 3.1 Final Model

Table 3.1: Variables used in final model

|  |  |  |  |
| --- | --- | --- | --- |
| variable | type | role | source |
| distance | numeric | predictor | original |
| avg\_hr | numeric | predictor | original |
| max\_hr | numeric | predictor | original |
| avg\_run\_cadence | numeric | predictor | original |
| max\_run\_cadence | numeric | predictor | original |
| total\_ascent | numeric | predictor | original |
| total\_decent | numeric | predictor | original |
| avg\_stride | numeric | predictor | original |
| min\_elevation | numeric | predictor | original |
| max\_elevation | numeric | predictor | original |
| best\_pace\_sec | numeric | predictor | original |
| aerobic\_TE | numeric | predictor | original |
| aerobic\_fct | nominal | predictor | original |
| anaerobic\_value | numeric | predictor | original |
| anaerobic\_fct | nominal | predictor | original |
| short\_distance | nominal | predictor | original |
| middle\_distance | nominal | predictor | original |
| long\_distance | nominal | predictor | original |
| rhr | numeric | predictor | original |
| avg\_pace\_sec | numeric | outcome | original |

The final model adopted for this project was a random forest which used 1000 decision trees, as well as 10-fold cross-validation. This resulted in a final RMSE value of 6.4. While this doesn’t meet the predefined benchmark of less than 5, this seems to be the best value achieved for a model that processes within a reasonable amount of time.

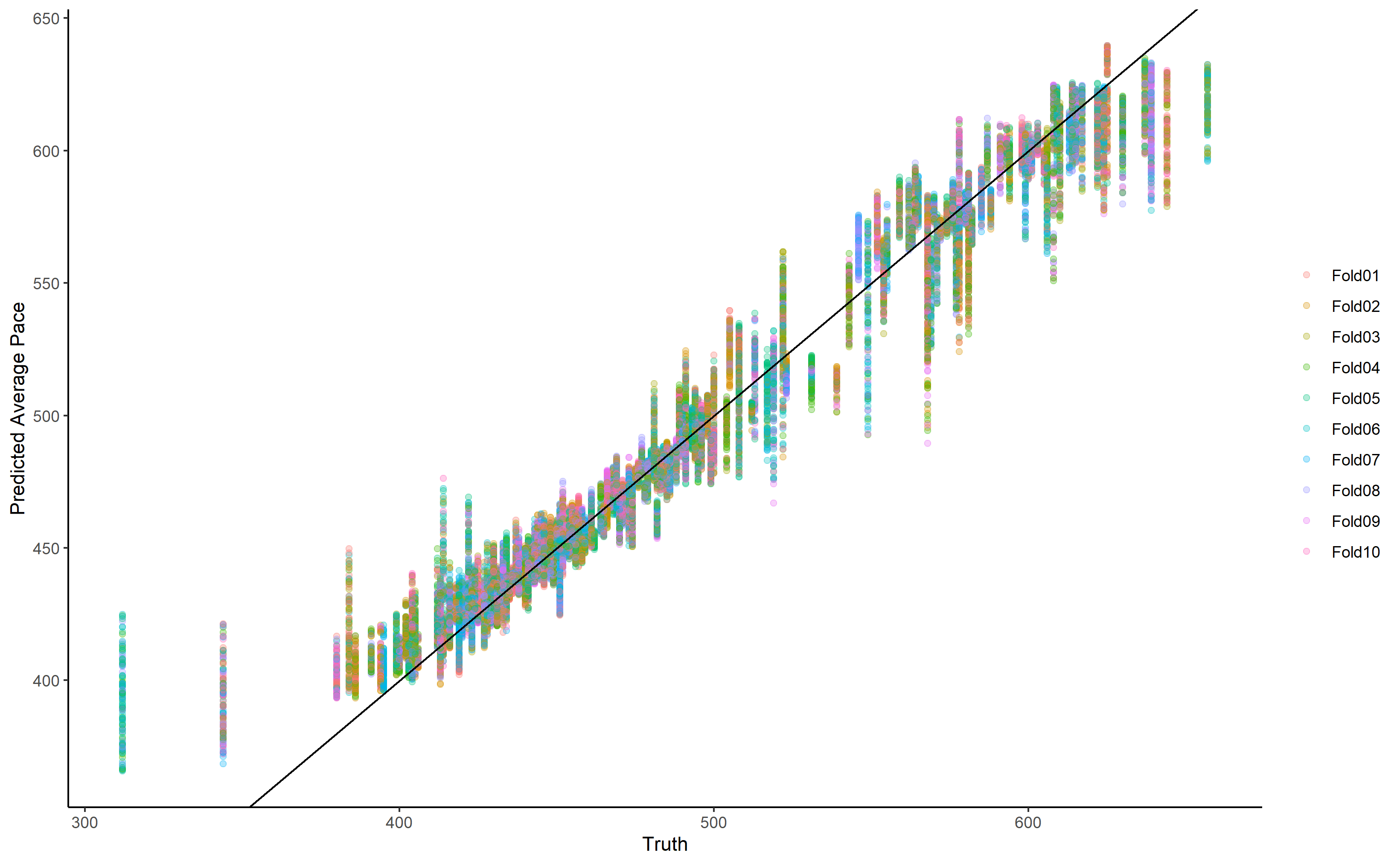


Figure 3.1: This busy figure shows how this model trained. Each color represents which fold a prediction came from. Each point corresponds to the true value on the x-axis and the predicted value on the y-axis. This plot seems ot show that my model predicted faster paces to be slower than they were, and slower paces to be faster than they were.

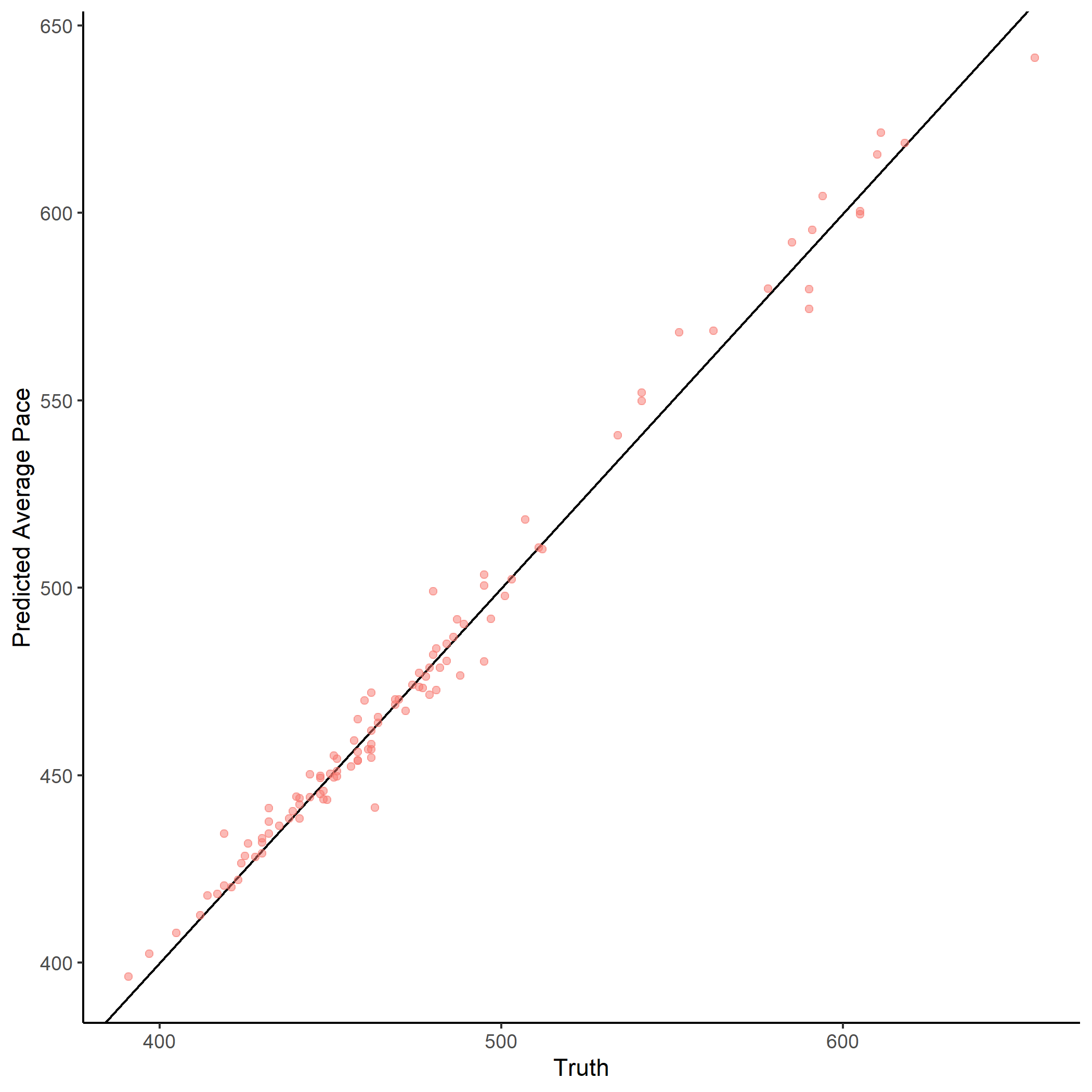


Figure 3.2: Final Predictions from random forest model. Comparing this figure to the model training figure, values tend to fall much closer to the line of best fit.

Table 3.2: Random Forest Results

|  |  |  |  |
| --- | --- | --- | --- |
| .metric | .estimator | .estimate | .config |
| rmse | standard | 6.3865715 | Preprocessor1\_Model1 |
| rsq | standard | 0.9867331 | Preprocessor1\_Model1 |

## 3.2 Other Models

Several more models were built during this project, including several random forests and a LASSO regression. However, these models were primarily used to better explore my data and tune my final models. More information about these models can be found in the supplementary material.

# 4 Discussion

## 4.1 Summary and Interpretation

In this project, I used data from a common smart watch to predict average pace for a run. The objective was to predict pace with an RMSE of 5 or less. While this was not accomplished, the final model was able to predict average pace with an RMS of about 6.4. Having the ability to predict average pace provides another useful data point for athletes in training - it gives a general sense of current race conditioning and current fitness. This model processes quickly enough that it can be used frequently with new data to provide updated information about how an athlete could perform in a 1-mile race, a 5k race, or a marathon on any given day. Such a data point provides good information for benchmarking in training plans, as well as planning and modifying activities to meet a goal.

## 4.2 Strengths and Limitations

### 4.2.1 Strengths

The greatest strength of this project is that most anyone with a smart watch and run data can repeat my work. While my project focused on using a Garmin Forerunner smart watch, other popular watch manufacturers provide users with similar data. For distance runners looking for this additional information - a race prediction time - they can plug in their own data to train a similar model. The resulting random forest model processes in a reasonable amount of time and provides a decent prediction.

### 4.2.2 Limitations

This model could be greatly enhanced with more data and more relevant features. However, this project proved that it was difficult to figure out which features were most relevant and which just caused more noise in the data. For example, I initially thought going through the trouble of having an additional variable for speed in miles per hour would have some useful application. Ultimately, it caused issues in my models.

I think my model would function even better with more observations. Coming into this project, I thought I had more useful data than I actually did, which limited my observations to just over 400. After another year of collecting observations, I think this model could be updated to provide more accurate predictions.

# 5 Conclusion

Average pace can be predicted accurately given the right variables. While my project was unsuccessful in meeting my self-imposed benchmark (RMSE < 5), it serves as a proof-of-concept for building predictive models for distance running. The best way to enhance such a model would be to add more data and verify whether features bring value to the resulting model.