Personalized Exercise Duration Prediction for Fitness Data

Ayushi Sharma	Soyon Kim	Shyama Shastri	Stefanie Dao
A15886693	A16121832	A15393859	A15974899
CSE 158	CSE 158	CSE 158	CSE 158
aysharma@ucsd.edu	sok020@ucsd.edu	sshastri@ucsd.edu	ctdao@ucsd.edu

ABSTRACT

Endomondo was a social network that allowed users to document individual physical activity sessions in order to help them track their progress and motivate them to improve their health. When logging workout sessions, users can include such details as what sport/activity they participated in as well as sensor data collected from various wearable devices, such as Fitbits or Apple Watches. In this project, we make use of a publicly available dataset containing Endomondo entries and continuous data sequences provided by such wearable devices during physical activity sessions. We then used this data to train a model to predict the duration of exercise. In order to improve the accuracy of workout duration prediction, we experimented with three different models and various subsets of preprocessed input features. After numerous attempts, we were able to train a random forest regression model that achieved a reasonable accuracy for our task. We conclude this paper with an analysis of our results and a brief overview of existing literature within similar problem domains.

Keywords

Fitness; Wearable sensors; Endomondo Dataset; Performance; Random Forest Regression; Linear Regression; Support Vector Regression

1. DATASET

1.1. Endomondo Dataset

We use the EndoMondo Fitness Tracking Data, which consists of basic contextual data such as gender, the sport they engage in, and the time elapsed since their last exercise and sequential sensory data such as heart rate, location (longitude, latitude, and altitude), and speed.

The dataset we use has discarded abnormal data such as those with overly large magnitude, mismatching timestamps, and abrupt changes in the GPS coordinates.

The dataset consists of 167,783 workout samples contributed by 1,059 unique users, each user having an average of over 175 workout records across the span of 2 years. Each workout sample is prefaced with a user ID, gender, sport type, and id. Each sample also contains sequences of latitudes, longitudes, altitudes, heart rates, and various derived sequences corresponding to a sequence of unix-formatted timestamps.

1.2. Exploratory Analysis

We began with the filtered version of the dataset, which consists of 167,783 workouts by 1059 users and a total of 27 unique sports.

The 10 most popular sports exercised by the users were:

```
[bike: 71915, run: 70591, mountain bike: 10722, bike (transport): 7757, indoor cycling: 1725, walk: 1289, orienteering: 867, cross-country skiing: 789, core stability training: 448, fitness walking: 292],
```

and the 10 least popular sports exercised by the users were:

```
[stair climbing: 3, pilates: 3, rugby: 3, yoga: 1, treadmill walking: 1, sailing: 1, kite surfing: 1, squash: 1, martial arts: 1, windsurfing: 1].
```

Of the 1059 users, 927 were male, 96 were female, and 15 were of unknown gender.

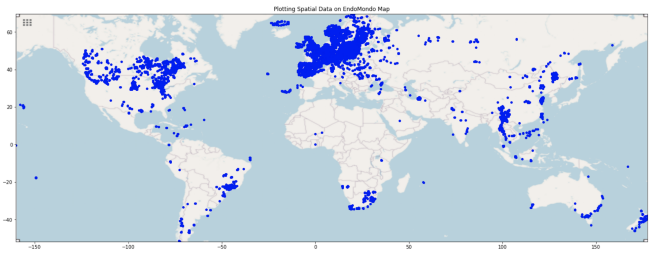

Figure 1. ***Geographic distribution of the users and their workout***

This plot shows the geographical distribution of the users and their workouts and demonstrates that a majority of users of the EndoMondo app are located in Europe and North America.

The following shows the average duration (in minutes) and average workout route (in meters) of the exercise for the top 10 most popular sports:

sport	average duration	workout route
bike	108.64	607.33
bike (transport)	60.36	288.74
run	69.18	343.93
mountain bike	116.74	832.86
orienteering	71.64	355.70
indoor cycling	73.67	162.58
cross-country skiing	107.25	766.83
core stability training	77.19	396.87
walk	69.05	236.58
fitness walking	82.76	380.81

Table 1. ***Average duration and workout route of the 10 most popular sports***

In Table 1, the greater the duration of the sport, the greater the workout route, demonstrating a strong positive correlation between the two variables. All of the workouts were longer than an hour, and the range of the workout route was from 162 to 832 meters.

2. PREDICTIVE TASK

We use the EndoMondo dataset to predict the performance, or average time duration, of a user's next exercise given a specific route (a sequence of longitude, latitude, altitude) and the sport they will engage in. This predictive task is a form of *regression*, in which our goal is to predict real-valued y (duration) as closely as possible based on predictor's variables x.

The forecast we sought to make could become helpful in helping users keep track of their personal fitness data and plan their future workouts more efficiently. Our purpose is to build a model that provides a personalized workout duration prediction based on a user's fitness profile.

2.1. Feature Selection

2.1.1. Data Processing

To get the most personalized prediction for a user, we first collect sports that have fewer than 50 workout records, then remove all users' workout records that are associated with these less popular sports. We then filter out users who have fewer than 10 workouts. After processing, our dataset consists of 160,053 records and 818 users, compared to 167,783 workout samples and 1,059 users as before.

We start with extracting the label that we want to predict by computing the duration of all records in our dataset. We do this by taking the difference between the last and first element in the *timestamp* sequence for each record, which indicates the time a workout starts and ends. We also converted this duration from seconds (Unix) to *minutes* as we found such unit changes would generate more meaningful and accurate results.

In addition, we also convert timestamp (Unix) to the standard date/time format and record which hour and day of the week each workout was recorded in our dataset.

As there are many sequential variables in our dataset, we need to use different methods to re-represent them before performing further analysis. For heart rate, derived speed, altitude, longitude, and latitude variables per a single workout, we represent them in

terms of maximum, minimum, and median values of each sequence.

We decide not to use the *distance* variable provided as we find it unreliable due to various negative magnitude values. Instead, we compute the 2D great-circle distance between 2 points using Vincenty's formula, then combine it with the difference in these altitudes to calculate the Euclidean distance between each pair of consecutive points in the longitude, latitude and altitude sequences provided. Finally, we sum up these small distances up to represent the total distance per a single workout record.

We also construct various data structures to record the number of workouts and the average workout duration per user, per particular sport, and per gender. For a single user, we also build their workout history with regards to the sport they engage in and sort their records in chronological order. These data structures can help us extract relevant information easily to select important features for our model.

2.1.2. Feature Analysis

As we have discovered in our exploratory analysis of the dataset, we can see the choice of workout route is strongly positively correlated to the duration. We further perform evaluations of each characteristic of a specific route against duration to test for correlation. We find that total distance and altitude have the greatest impact on the exercise duration. Therefore, we select distance and altitude to be our features to represent a route for prediction.

User's profile is another important factor that we found to have a strong correlation with duration. Therefore, we extract data related to a user's workout profile, such as average workout time, to build our feature vector. In addition, we observe that the exercise duration of most users seems to be similar to their previous workouts of the same sport (Figure 2). We also account for this in our prediction's feature representation by representing a user's average time of most recent workouts of the same sport (the number of workouts will be adjusted) based on the number of workouts users have done for that sport).

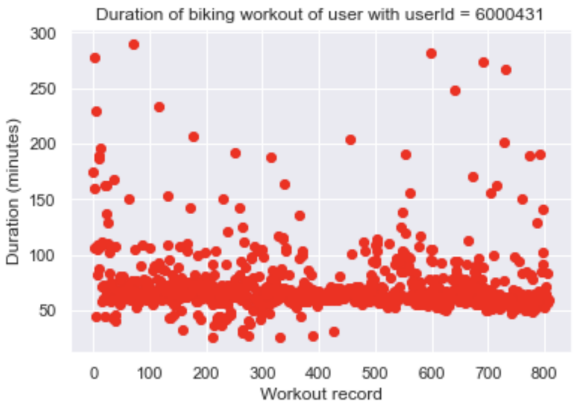

Figure 2. *Duration of biking workout* for a specific user

Other important variables that we found in the previous exploratory analysis are gender, speed, and time since the last workout (since_last). We decide to leave out other variables such as heart rate, time_elapsed (since there is a lot of noisy data that reduces the accuracy of our prediction).

During our model training phase, we will also experiment with different combination of features as well as variations of each feature in order to design the best set of features for a specified model.

Feature Name	Encoding
userId	int
distance	float
gender	one-hot
speed	float
day_of_the_week	one-hot
avg_workout_duration_history	float
distance	float
max_altitude	float
since_last	float

Table 2: *List of features used in the model and their encoding method*

3. MODEL SELECTION

3.1.1. Machine Learning Model

All models used variations of the following feature vector that we described in the previous section. The output label common to all models was a single float, representing the predicted duration (*minutes*) of activity.

We split the dataset into a training set (75%), a validation set (12.5%), and a testing set (12.5%). We first trained the model on training set, validate it on validation set and test its performance on test set.

We attempted to use four different models:

a. Baseline

We first identify a baseline to compare our models. Our simple baseline is a model that always predicts the average workout time of all workout records in the training set.

b. Linear Regression

As our prediction is a regression task, we will select a regression model that can help us predict real-value labels as closely as possible. Our first attempt involved a standard linear regression model. Later, we strive to experiment with other models to improve our performance compared to the Linear Regression model.

We attempted to improve this model's performance by experimenting with different sets of features. We found our Linear Regression performs best when we replace average speed with max speed, encode the day of the week feature, and add a more personalized feature such as the user's average workout duration of the same sport.

Feature Representation	MSE
userId, sport, gender, average speed, distance, max altitude	2353.1646
userId, sport, gender, max speed, distance, max altitude, day of the week	1952.8157
userId, sport, gender, max speed, distance, max altitude, day of the week, previous workout duration, since_last	1731.1086

Table 3: ***Comparison of linear regression performance for three different feature subsets***

c. Support Vector Regression

After Linear Regression, another regression model that we decided to try out was Support Vector Regression (SVR). This has the same working principles as the Support Vector Machine but for the regression model [5]. SVR allows us to model non-linear relationships between variables (which is the case of our dataset) and provides the flexibility to adjust the model's robustness by tuning hyperparameters.

However, we quickly realized this type of model does not work really well with large datasets (especially datasets with sizes larger than 10,000). As a result, we obtained a very huge MSE at test time (around 73770946.38103).

d. Random Forest

Random Forest is an ensemble model that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. We choose this model to experiment with because it usually works well on large datasets and for non-linear relationships. Since our dataset consists of more than 100,000 records and there is no linear pattern between input features and labels, we think Random Forest could give us more accurate predictions.

For this model, we also tried different representations and found that the same set of features that work best for linear regression also perform best for the Random Forest model. We also attempt to replace speed average with speed median, add user's average workout duration feature, and tune model's parameters which we found to have improved our prediction significantly. Even though this is a complex model, we did not run into overfitting issue during our experiment.

We also plot our model prediction and the actual workout duration for different users to visualize how accurate our model can predict (Figure 3).

Figure 3: ***Comparison of the actual duration of workout vs. the Random Forest prediction during test time for the first 50 records***

We also visualize the top important features for our model. From our observations, distance and workout time history (average duration of user's previous workouts of the same sport) contributes most to our prediction.

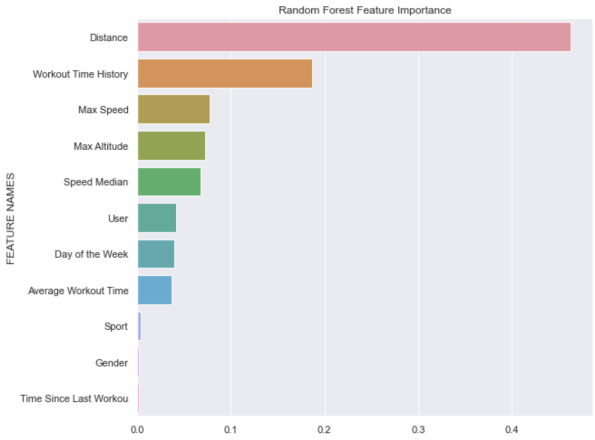

Figure 4: ***The weights of individual features in the random forest regression model***

3.1.2. Model Evaluation

After creating a training dataset using 75% of the entries in the total dataset, we will create a test dataset using 12.5% of the total dataset in order to evaluate our model. The validation dataset, which takes the remaining 12.5% of the total dataset, is evaluated using mean squared error (MSE) and R^2 score. We then calculate the MSE between the predicted results and test values and compare the MSE across different models. We also use the R^2 coefficients as another way to evaluate our models.

4. LITERATURE

We are using an existing dataset for our prediction task. Our data is a collection of workout logs for different users and their workouts are tracked using Endomondo.

This data was used as a part of an existing project: FitRec-Project. This project utilized 3 different versions of the EndoMondo Fitness Tracking Data. The first version was just the original raw data that was recorded from the Endomondo tracker app. Another version of the preprocessed data was sampled in such a way that it would be beneficial for making predictions for a short-term heart rate prediction task. Lastly, the third version of the data used in this project was the one that they used for predicting routes for a given user's workout [4]. This last version of the data is what our team decided to use as well.

The FitRec-Project proposes an LSTM-based model that predicts the user's heart rate for a given sport and recommends the most suitable activity accordingly. While this project studies a different research question from ours, both projects demonstrate a model that is context-aware and personalized to specific users.

For our prediction task, our team decided to use the last version, firstly, because it filters out a lot of abnormal data points. It also does additional preprocessing, such as taking out users with less than 10 workouts. It is also not set up in a way to make short-term predictions like the 3rd version of the original dataset, another reason we decided to use this specific dataset. As a result, we used the second version of the data and then preprocessed it even further as needed for feature extraction and the process of trying out different models.

As mentioned earlier, this dataset that our team decided to use was also used for another predictive task that involved suggesting routes to a user given "an expected workout time, an idiosyncratic heart rate or speed curve" [1]. A user's recent workout sequences are used to forecast their future workout activities. This project made these predictions using the LSTM model.

In attempts to find similar datasets, we found that not a whole lot of people are using fitness data from trackers like Endomondo for making recommendations and predicting stuff based on users' past activities. As a result, we were not able to find any similar datasets that held information about users in such a manner that we got from the Endomondo dataset. This made it harder for us as we could not compare our results to anything outside as there were not any existing predictive tasks that involved using such kinds of datasets to make predictions about time.

Even though we were not able to find similar datasets, we found projects that involved predicting the duration of a specific task instead. Even though the models that were used in these projects were outside the scope of what we have learned in this course, it

was still helpful to see similar predictive tasks and the way people were using different models to beat the baseline, how they were evaluating the accuracy, etc. We did find some predictive tasks that involved making predictions about time duration, such as how long it would take to finish a task or how long a user would take to complete the task. One such project, we found, was used for "predicting execution time for spark jobs" [3]. In this paper, it is described how the team used built-in MATLAB packaging models and models like MARS and NNLS to complete the predictive task. For validating the accuracy of their models they also used R^2 and MSE. Their evaluation methods are similar to ours as we also relied on finding MSE and R^2 to evaluate the performance of the models that we experimented with.

Other literature relevant to the topic of predicting the time duration of an exercise includes existing projects using information from the dataset that is showcasing results from a tracker like Endomondo, Fitbit, Apple Watch, etc. One such project that we found used a deep learning model to evaluate the "physiological representations from large-scale wearable data" [2].

Overall, previous literature shows that the data collected from wearable devices and apps with sensors such as Endomondo have been utilized to predict the user's performance (e.g. heart rate explained in) and physical conditions [1, 2]. While our project shares the same dataset with some of the previous literature, our project expands on them and explores the personalized prediction of a specific user's average duration of an exercise.

5. RESULTS

Model	MSE	R^2
Baseline	3207.9055	0.24
Linear Regression	1731.10864	0.485
SVR	73770946.38103	-
Random Forest	887.16932	0.72343

Table 4: ***Comparison of the best performance of various implemented regression models***

Overall, except for SVR model, both Linear Regression and Random Forest Model performed well in both training and testing time, with no issue of overfitting. In comparison, the SVR models performed extremely poorly, with an MSE of 73,770,946 at test time. We saw the greatest improvement in the Random Forest Model, with an MSE of approximately 887, nearly half of the MSE of the Linear Regression Model, and a quarter of the MSE of the baseline model (always predicting the average duration). We, therefore, have demonstrated a model that meaningfully predicts the user's duration of a candidate exercise.

We found that the distance was by far the most important feature for predicting workout time in our Random Forest Model, with a weight nearly double that of the next most important feature, workout time history. This is reasonable as any distance traveled can involve a predictable amount of time taken. The significance of the next most important feature, workout time history, can be explained by the fact that users are likely to engage in physical activity for designated amounts of time per somewhat rigid routines.

Unexpectedly, features that we assumed would work well such as gender or sport type do not contribute much to our predictions. We can assume that gender information does not provide much helpful information such as a user's physical characters, which could have better our prediction if given. Sport type does not seem to be a determining factor either, since users can spend a lot of time for any sport to their preferences.

Our best-performing model, the Random Forest Regression Model, has two parameters that we tuned to improve the performance: the *n_estimators* and *random_state*. The *n_estimators* determine the number of trees in the forest model. The *random_state* controls the randomness of the bootstrapping of samples and the sampling of features.

Parameters	MSE
$n\_estimators = 10$, $random\_state = 0$	1278.92385
$n\_estimators = 50$, $random\_state = 0$	991.52347
$n\_estimators = 100$, $random\_state = 0$	894.43014
$n\_estimators = 500$, $random\_state = 10$	887.16932
$n\_estimators = 500$, $random\_state = 20$	899.14196

Table 5. ***MSE for Random Forest Model with tuning parameters***

For our Random Forest Model, we find that setting our parameter *n_estimators = 500, random_state = 10* gives us a better prediction with the smallest MSE during testing time.

Our best model, Random Forest, succeeded primarily because it works best for dataset that have large size (our dataset has around 160,000 points) and non-linear relationships between input features and the target variables. Our linear regression model has decent performance as it even outperforms the MSE we got from testing out the SVR model. SVR performed the worst and we think primarily because it is a general trend for SVR to perform poorly on huge datasets.

6. REFERENCES AND CITATIONS

[1] Jianmo Ni, Larry Muhlstein, Julian McAuley, "Modeling heart rate and activity data for personalized fitness recommendation", in Proc. of the 2019 World Wide Web Conference (WWW'19), San Francisco, US, May. 2019.

[2] Spathis, Dimitris, et al. "Learning Generalizable Physiological Representations from Large-scale Wearable Data." arXiv preprint arXiv:2011.04601 (2020).

[3] Mustafa, Sara, Iman Elghandour, and Mohamed A. Ismail. "A machine learning approach for predicting execution time of spark jobs." Alexandria engineering journal 57.4 (2018): 3767-3778.

[4]https://sites.google.com/eng.ucsd.edu/fitrec-project/home

[5]https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/