Public Data Science FitnessKeeper Final Report

Julian Morris and Nitya Dhanushkodi

Note: This report has had figures removed so it could be shared publicly.

Table of contents:

I.Overview and Goals

The goal of this project is to try to understand fitness routines of the users of RunKeeper, a fitness app by the company FitnessKeeper. As a part of this, the goal is to try to predict when someone might lapse in their fitness routine based on the data. Another goal with this dataset is to predict when someone will buy the elite version of the application. In predicting lapsing and whether or not someone has the elite version of the app, we try to quantify and visualize routines to understand what kind of routines people who lapsed and people who didn't had.

II. Data Exploration

The data given recorded workouts starting in April 2014 to six months later. The criteria for users to be in the dataset is that they had to have one workout before April 2014 and have one workout in the timeline of April 2014-October 2014.

First, we look at individual variable distributions in the data to see what the data is like and what relationships seem important to explore, while also keeping the goals of the project in mind. The variables in the dataset are as follows:

'tripid': is a unique id number for each fitness activity logged**.**
'platform': is Android, iPhone, web, other
'entry_type': GPS or manually entered

'starttime': local start time of trip
'distance_mi': trip distance in miles
'duration_min': trip duration in minutes
'userid': User ID
'gender': M or F gender of user
'age': The user's age
'region': (CT, VT, NH, MA, RI, or ME)
'creationdate': Runkeeper account creation date
'trips_before': Number of running trips recorded before April 2014
'distance_before_mi': Total distance for activities recorded before April 2014
'duration_before_min': Total time for activities recorded before April 2014
'is_elite': Whether the user is a subscriber to a publication or not (1 or 0)
'genderNum': gender of user mapped to a number (F:0, M:1)
'pace': duration_min/distance_mi to get a pace in minutes per mile

We also created some new variables to better analyze the data:

deltaDays: The number of days between each workout.

stdDelta: standard deviation of days between each workout. The higher the stdDelta, the more erratic the workout schedule. We used this as a way to quantify routine

pace: duration/distance in minutes per mile

lapser: labels a lapser with a '1', and a non-lapser with a '0'. A user that has any instance of deltaDays of more than 21 days is a lapser.

Cleaning

We clean the data to include only the workouts that were between 0.5 and 30 miles, 10 to 360 minutes long, and pace between 3 and 30 minutes per mile. This is done because the user could have turned the app on by accident, resulting in a workout logged for less than 0.5 miles or less than 10 minutes. We also excluded extremely high durations, distances, and fast paces that seemed impossible for a workout.

Before cleaning	After cleaning
341,986 workouts	325,145 workouts
10,000 users	9,994 users

Figure 2-1: Our data before cleaning and after cleaning.

Demographic

We looked the histogram of ages and proportions of gender to look at the demographic of our user group. The mean age of the users is 35.9 years. Most of this data comes from an adult user group.

Female	Male
5600	4400

Figure 2-2: Number of male and female runners in the dataset.

There are 10000 users in the data set, and according to figure 3, 44.0% are male and 56.0% are female users. Overall, our dataset is based off a more female than male and to a mostly adult group. It is a little unusual that there are so many more women than men in this dataset, even though the US has slightly more women than men. Perhaps it is because this dataset only has users that chose to identify their gender in the first place, which may have resulted in a skew.

User Averages

We show histograms of the average distance, duration, and pace of each user in the dataset to see generally how far and for how long people are working out. These variables may be important when predicting whether users will lapse in their fitness routine or if they buy elite, so it could be valuable to get an intuition for the distribution.

Most users seem to have average workout distances between 3 to 5 miles. In fact, the average workout is 3.86 miles across all workouts.

Most users seem to work out on average from 20 to 50 minutes long. The average duration for workouts is 39.6 minutes across all workouts.

About half the users seem to run between 9 and 11 minute miles. Almost all users run between 5 and 20 minute miles. Workouts with paces greater than 30 minutes per mile are excluded because the user may have accidentally turned on the app while walking around and not actually working out.

Workout History

We are also given past workout data from before April 2014: the number of workouts before April 2014, total distance in workouts before April 2014, and total duration in workouts before April 2014. To see how many users have been regularly using the app, look at histogram of number of workouts before April 2014 across the users.

About 40% of the users have very few workouts before April 2014, indicating they did not build up much of a workout history. This variable is important to look at, because for the 60% of users that do have trips before April 2014, we think their workout history may have important implications for whether the user may lapse.

Proportion of Elite Subscribers and Lapsers

The benchmark for determining prediction accuracy for elite status and Elite status is about whether or not a person has bought the elite version of the app.

Elite Subscriber	non-Subscriber
1815 (18.2%)	8179 (81.8%)

Figure 2-8: Elite subscribers, where 1 is an elite subscriber, and 0 is a non-subscriber across the 9994 users remaining after cleaning.

In the dataset, 18% of the users have the elite version of the app, and 82% don't have elite. This is useful to compare to the logistic regression model prediction run in the logistic regression section, so we know how accurate the logistic regression model will be even if it predicts 'not elite' or 0, every single time.

We have defined lapser to be a user that has not worked out for 3 weeks straight or longer at any point during the 6 month period that we have data for.

Lapser	non-Lapser
6053 (60.6%)	3941 (39.4%)

Figure 2-9: Lapser information, where 1 is a lapser, and 0 is a non-lapser across the 9994 users.

In the dataset, 60.6% of the users are lapsers, and 39.4% are not. These numbers will also be used as a benchmark to compare the performance of our classification performance.

Relationships Between Variables Related to Routines:

	is_elite	stdDelta	lapser
distance_mi	0.130413	-0.194992	-0.171985
duration_min	0.119340	-0.158708	-0.141533
age	0.083101	-0.134954	-0.128762
trips_before	0.238930	-0.203794	-0.192732
distance_before_mi	0.222187	-0.198654	-0.187213
duration_before_min	0.231176	-0.197907	-0.187587
is_elite	1.000000	-0.114532	-0.101054
pace	-0.036268	0.093225	0.067491
genderNum	0.137727	-0.014093	-0.013644
stdDelta	-0.114532	1.000000	0.473104
lapser	-0.101054	0.473104	1.000000

Figure 2-12: Correlation table between variables, and the three more important ones- elite status, stdDelta, and lapser.

It seems that the longer distances, longer durations, more trips before April 2014, longer distances and durations before April 2014, and having the elite version of the app is correlated with not lapsing. The most correlated variable with lapsing, is stdDelta. A high stdDelta is correlated with being a lapser.

stdDelta is the standard deviation of the number of days in between each workout for a user. A high stdDelta indicates very different amounts of time in between workouts, and a low stdDelta indicates a consistent amount of time in between workouts. At this point, the stdDelta variable will definitely be influenced by a lapse (determined to be 3 weeks or more without working out) and a next iteration of quantifying routines could involve not including the lapse as part of the routine, and use the routine as a predictive variable.

The most highly correlated variables that aren't for obvious reasons look like something we can use later in predicting what we want to. A correlation found not in this table is distance_before_mi is correlated with distance_mi with a correlation coefficient of 0.356.

This means that these two variables might be interesting to look at to see how much the user ran before can be used to predict future fitness routines.

For the is_elite variable, it was surprising to find that not many variables were strongly correlated with it. The trips_before/distance_before_mi/duration_before_mi, variables had the strongest correlation of around 0.23. Data before 2014 seems to have the greatest predictive ability in whether a person subscribes to the elite membership. Perhaps this is indicative of our results in the logistic regression section to predict is_elite and the ML algorithm section.

III. Time series Routine Analysis:

To visualize and quantify routines, we use two methods: a normalized graph of user routines to allow for comparison and quantification, and autocorrelation. In this section, we describe the normalized routine. For each user, we input 1 for the days they have worked out, and 0 for days they have not. By normalizing across the routines using a CDF, we can look at user routines and compare them to eachother and to a daily routine. The CDF also allows us to compare users with a different number of workouts.

To interpret the CDF graph, look for how close a user is to a line drawn diagonally from their first workout to their last. This line would represent a daily routine. Deviations from this show irregularity. Horizontal lines indicate a period of time gone without the user having worked out.

This allows us to visualize user routines in a normalized form. Since each user may have logged a different number of workouts, this allows us to compare user routines on a normalized scale.

For now, these graphs serve as a good way to compare routines between users visually. We can also see these being used to compute some sort of quantitative coefficient to measure how regular of a routine each user has. Since a ‘perfect’ daily routine would result in a straight line from the starting day to the ending day, we take the area between that straight line for a particular user, and their actual CDF routine. We can then characterize how far they are from the ‘perfect’ routine and quantify how much they stick to their routine. While we did not choose to go in this direction and quantify the routine this way, this spurred another idea.

We chose to create a variable, stdDelta, which is the standard deviation of the distribution of days between each workout for a user. A high stdDelta would indicate

high irregularity in the number of days between workouts and a low stdDelta would indicate a consistent number of days between workouts.

IV. Autocorrelation:

The other direction we took to quantify routines was to look at periodicity in routines using autocorrelation. We chose not to use this as a technique to quantify routines.

We put in an array of durations of the workouts for each user if they worked out that day. If a user worked out on day 0,2,5,6,7,and 9, for 30 minutes on each day, the input array for the autocorrelation function would be 30,0,30,0,0,30,30,30,0,30. Here are the results for some different types of users:

Plotting autocorrelation and workout durations over time is helpful in visualizing what a regular routine, and a not so regular routine looks like. It's a little more difficult to compare autocorrelation functions between users, other than when there is a very rigid routine, actually causing the correlation to peak a significant amount. These three users are very different in how much of a routine they have, but it is hard to tease apart what exactly to quantify differently from their three autocorrelation functions.

The workout duration graph is useful in quickly visualizing lapses, and could be useful in characterizing a more personal definition of a lapse. We very simply define a lapse as not working out for longer than 3 weeks straight. However, from these workout duration graphs, you could calculate the average time between workouts for a particular user and characterize a lapse for that user from that, or simply characterize a lapse visually.

Unfortunately, since it is so hard pick out a feature in the autocorrelation function to differentiate between someone who has a semi-regular schedule and a very irregular schedule, we decided not to use this as a technique to quantify routines.

V. An In-Depth Comparison between Lapsers and Non-Lapsers

In order to compare some of the features given to us for lapsers vs. non-lapsers, Figures 5-1, 5-2, and 5-3 are some statistics on lapser and non lapser users. The rest of the figures are cdfs of lapsers and non lapsers. This gives an idea of what variables might have an influence in predicting lapsing.

Lapser	non-Lapser
6053 (60.6%)	3941 (39.4%)

Figure 5-1: Number of users that are lapsers and non-lapsers

	male	female
Lapser	2627 (43.4%)	3426 (56.6%)
not lapser	1765 (44.8%)	2176 (55.2%)

Figure 5-2: A table of male/female lapser/not lapsers.

Note that a lower proportion of lapsers are male than not lapsers. This could indicate that a male is more likely to not be a lapser than you would otherwise expect, because the proportion of male not lapsers is higher than 44%, which is the proportion of male users in the entire data set. However, the difference is small, so gender may not be a great predictor of lapsers.

	elite	not elite
Lapser	909 (15.0%)	5144 (85.0%)
not lapser	906 (23.0%)	3035 (77.0%)

Figure 5-3: A table of elite/not elite lapser/not lapsers.

Note that a lower proportion of lapsers are elite than not lapsers. An elite person is more likely to not be a lapser than you would otherwise expect, because the proportion of elite not lapsers is higher than 18%, which is the proportion of elite users in the entire data set. The difference is fairly large, so elite status may be a good predictor of lapsers.

The average distance is higher overall than lapsers, which is interesting and may be useful to use as a variable to predict lapsers. The users who lapse have a slightly slower pace than not lapsers, but not by much.

The average age is higher overall for non lapsers, which is interesting and may be useful to use as a variable to predict lapsers. The lapsers have less trips before than not lapsers. It is interesting that not lapsers seem to have more past workouts than lapsers.

People who are not lapsers have a significantly lower stdDelta than lapsers, meaning that lapsers have a more regular routine than not lapsers. stdDelta indicates how variable the time in between workouts is, and a high stdDelta indicates a very erratic workout schedule. It seems that lapsers do have a more erratic schedule than non-lapsers. Since there is a significant difference in the users, stdDelta will most likely be a good predictor of lapsers.

VI. Predicting elite status:

With an intuition for what variables may have an influence in predicting elite status, we use Logistic Regression to predict this. The results are shown below:

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
Intercept	-3.0292	0.235	-12.902	0.000	-3.489 -2.569
distance_mi	0.1721	0.023	7.534	0.000	0.127 0.217
trips_before	0.0063	0.001	6.871	0.000	0.004 0.008
distance_before_mi	-0.0020	0.000	-5.579	0.000	-0.003 -0.001
duration_before_min	0.0002	4.06e-05	3.977	0.000	8.19e-05 0.000
pace	0.0588	0.015	3.859	0.000	0.029 0.089
genderNum	0.7011	0.060	11.643	0.000	0.583 0.819
lapser	-0.1390	0.066	-2.107	0.035	-0.268 -0.010
stdDelta	-0.0128	0.003	-4.810	0.000	-0.018 -0.008

Figure 6-1: Results from Logistic Regression model to predict is_elite

Large coefficients indicate that the variable on the left has a large effect in predicting the status of is_elite. Low p-values indicate that it is very likely to see a coefficient as large as the coefficient is. All of these variables seem statistically significant, with genderNum,

distance_mi, and lapser having the largest effect on the outcome, but they are not very good at predicting whether or not the user buys elite.

We split our data into test and train sets, use the Logistic Regression module from scikit learn on the training data to train the model, and use this model on the testing data. By comparing the prediction for the test data with the actual values for is_elite in the test data, we see how accurate the model is in predicting is_elite status. It is 82.2% accurate in the predictions.

However the proportion of elite members is 81.8%, and if the model simply predicts false all the time, it can achieve 81.8% accuracy. The predictions of the model do show that it predicts false all the time. The value of this model is to show what variables might have some predictive significance in machine learning algorithms. Given this information, we will likely use most of these variables in the machine learning algorithms.

VII. Predicting Lapsing:

We label each user with a lapser category giving them a ‘1’ if they are a lapser and a ‘0’ if they are not. We defined a lapser to be a user that has lapsed at least once, where a lapse is a time when a person has not worked out for longer than 3 weeks straight.

Our process was to identify the user id’s of lapsers and not lapsers, mark them as 1s and 0s in a separate dataframe, and merge it with the original dataframe to add on the extra lapser category to the entire dataframe. We then grouped by userid and took the mean of the groupby object to return a dataframe of averages of the quantitative variables for each user. We split our data into test and train sets, train the model, and use this model on the testing data.

The results are as follows:

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
Intercept	-1.6394	0.202	-8.097	0.000	-2.036 -1.243
distance_mi	-0.1360	0.021	-6.415	0.000	-0.178 -0.094
trips_before	-0.0037	0.001	-4.176	0.000	-0.005 -0.002
distance_before_mi	0.0003	0.000	1.818	0.069	-2.62e-05 0.001
pace	-0.0064	0.013	-0.495	0.620	-0.032 0.019
genderNum	0.0615	0.053	1.162	0.245	-0.042 0.165
is_elite	-0.1030	0.065	-1.577	0.115	-0.231 0.025
stdDelta	0.0964	0.003	37.375	0.000	0.091 0.101

Figure 7-1: The results from predicting lapsing using logistic regression. We'll focus on the coefficients for each of these variables and their influence in predicting lapsing.

The model predicted with 63.4% accuracy whether or not a user was a lapser based on the variables shown in the table.

The results show that the variables with the strongest relationship in influencing a prediction for a lapser or not is is_elite, and distance_mi, and stdDelta of workouts. Since having elite is labeled '1', having the elite version of the app will make the model less likely to predict a lapse, and having longer distances will make the model less likely to predict a lapse, and having more variability in when you workout, which is indicated by stdDelta, will result in the model more likely to predict a lapse.

Based on an understanding of the variables from data exploration, we also show a Logistic Regression model with the variables we expected to be more predictive. However, the result here was not too different in accuracy, so we may need more powerful algorithms described in the ML section.

Here are the results:

	coef	std err	z	P>|z|	[95.0% Conf. Int.]
Intercept	-1.7863	0.106	-16.875	0.000	-1.994 -1.579
distance_mi	-0.1181	0.018	-6.429	0.000	-0.154 -0.082
is_elite	-0.1077	0.065	-1.651	0.099	-0.235 0.020
genderNum	0.0676	0.051	1.323	0.186	-0.033 0.168
stdDelta	0.0964	0.003	37.397	0.000	0.091 0.101
trips_before	-0.0022	0.000	-7.072	0.000	-0.003 -0.002

Figure 7-2: Results of Logistic Regression model with seemingly more predictive variables.

The model predicted with 62.1% accuracy whether or not a user is a lapser. Like the previous case of predicting lapsing, this accuracy percentage is too close to the proportion of lapsers, 60.6%, so this model is mostly predicting true all the time.

The value of these logistic regression models is to give an idea of what variables might be most useful to put into more powerful algorithms. For lapsing, this is likely to be stdDelta, is_elite, and distance_mi.

VIII. Predictions using other ML algorithms:

Other machine learning algorithms may offer an explanation for the factors involved in predicting lapsing, and we found higher accuracies for lapsing prediction, but not for elite_subscription prediction.

Here is a list of other ML algorithms we tried using:

- Random Forest Classifier
- Adaboost Classifier
- Support Vector Machine (Linear and nonlinear kernels)
- Bagging Classifier
- Extra Trees Classifier
- Gradient Boosting Classifier

	Lapsing Prediction Accuracy	Elite Prediction Accuracy

Benchmark for comparison	60.6%	81.8%
Random Forest Classifier	72.8%	82.2%
Adaboost Classifier	76.8%	81.5%
Gradient Boosting	77.0%	81.4%
Logistic Regression	76.9%	81.7%
Bagging Classifier	71.2%	80.1%
Extra Trees Classifier	72.1%	81.0%

Figure 8-1: Prediction accuracies of different algorithms on lapsing prediction and elite prediction. Gradient Boosting performed significantly better than the benchmark accuracy at 77.0%. For elite prediction, only Random Forest Classifier performed better than the benchmark at 82.2%.

The variables used in these machine learning algorithms are distance_mi, age, trips_before, distance_before, is_elite, pace, genderNum, stdDelta, and lapser, although is_elite was not used in the prediction for is_elite and lapser was not used in the prediction for lapser.

It seems that the variable with the most predictive value for lapsing prediction is stdDelta. When you remove 1 or 2 other variables from the ML algorithms for lapsing prediction, the accuracy does not change that much. However, when you remove stdDelta, the percentage accuracy goes down to about 63-65% amongst the algorithms. So, doing further work in classifying routines in a quantitative way to use in a prediction model for lapsing is useful.

For predicting whether or not the user buys the elite version of the app, we find that when you remove any of the variable inputs to the ML algorithm, it doesn't affect the accuracy very much. The accuracy is still very close to the benchmark, and we suspect the models are likely predicting false for is_elite always.

IX. Other Exploration Results:

Hypothesis testing of difference in distance between iphone and Android users:

We ran a hypothesis test to see whether there was a statistical difference between the average distance run by iphone users and Android users. This is possibly an interesting side result but not as closely related to the goals of the project.

The test statistic was the difference in means between iphone and Android users. The mean distance of Android users is 3.88 miles and the mean distance of iphone users is 3.97 miles. The difference between these means is 0.0853 miles, and we want to see if this is statistically significance.

So, our null hypothesis is that there is no significant difference in average distance between iphone and Android users.After we ran the hypothesis test with the null hypothesis, the program outputted the p-value to be 0. Since our data consisted of 10,000 user data points, it means that the p-value is at least less than 0.001.
p-value < 0.001

In conclusion, here is a statistically significant difference in the mean distance travelled between iphone and Android users. Specifically, iphone users travelled an average of 0.0853 miles(137 meters) more than Android users. The reason for this difference is unknown, but may be iphone users have a greater family income than Android users, and maybe family income is correlated with how much they exercise.

Differences between states

Here are the average miles in a workout per state.
CT = 4.036
RI = 4.028
MA = 4.016
ME = 3.984
VT =3.976
NH = 3.943

As shown, the people in Connecticut seem to run longer distance workouts on average by 0.093 miles more than New Hampshire people.

X. Conclusion/Future works:

Overall, we found that lapsers were easier to predict than elite subscribers, and routines were difficult to visualize with autocorrelation, but easy to quantify with standard deviation of time between workouts.

For future works, there are several things we can do to improve our prediction model and understand routines better. To understand routines better, we can use different definitions of lapsers. One example would be to define lapsers based on the the individual's workout schedule so that a lapse is personalized to the user. This could be achieved by taking the mean of the days between workouts for a user and say that

anything outside of a 95% confidence interval from that mean is a lapse. Another way to characterize routine is to use the area under the curve in figure 3-1 as a variable. Also to improve routine characterization as a predictive variable in ML algorithms, we could make the routine characterization go up until the lapse but not include it. Right now, stdDelta includes the lapse itself, but if it were to go up until the lapse to help predict the lapse, that would be more useful to characterize.

To create a better prediction model, we should use GridSearchCV on multiple ML techniques. This will allow us to tune different parameters quickly for better results. We also did not use cross validation for our data or test our model with other data sets, so that will also be an important next step. Finally, we could use a different metric for accuracy in our model. For example, we could reward a correct label of a lapser or elite more highly than we do punish an incorrect label of a non-lapser as a lapser.

We enjoyed working with this dataset, and we learned a lot about what working with data is like. It was difficult to know what we should try next sometimes, but it was a very enjoyable process where we found (and didn't!) relationships and interpreted this dataset.