Recommendation in Persuasive eHealth Systems: an Effective Strategy to Spot Users' Losing Motivation to Exercise

Paolo Pilloni, Luca Piras
Dip.to di Matematica e Informatica
Università di Cagliari
Via Ospedale 72
Cagliari, Italy 09124
{paolo.pilloni,lucapiras}@unica.it

Ludovico Boratto
Digital Humanities
EURECAT
Camí Antic de València, 54
Barcelona, Spain 08005
ludovico.boratto@acm.org

Salvatore Carta, Gianni Fenu, Fabrizio Mulas
Dip.to di Matematica e Informatica
Università di Cagliari
Via Ospedale 72
Cagliari, Italy 09124
{salvatore,fenu,fabrizio.mulas}@unica.it

ABSTRACT

Persuasive health technologies in the sport domain focus mostly on motivating and supporting people in reaching an active lifestyle. In this paper, we exploit a real-world dataset made available by a commercial persuasive ecosystem called *u4fit*. u4fit allows coaches to create tailored workout plans and to constantly monitor and support their sportsmen remotely. Occasional sportsmen often and suddenly abandon their workout routines, without giving any prior notice to the coach, frequently because of a decline in motivation. In this paper, we tackle this issue by developing an approach able to spot users' behavioral changes and predict if one will soon stop exercising. These predictions can be further elaborated and provided as a recommendation to the user's coach, to let her get in touch with the sportsmen and prevent such a situation. Experiments, validated through standard accuracy metrics, revealed that behavioral changes in training patterns represent one of the main markers that lead sportsmen to abandon.

CCS CONCEPTS

• **Information systems → Mobile information processing systems**; **Data mining**;

KEYWORDS

Personalized Persuasive Technologies, Health Recommendation, Healthy Lifestyle, eCoaching, Motivation.

ACM Reference format:
Paolo Pilloni, Luca Piras, Ludovico Boratto, and Salvatore Carta, Gianni Fenu, Fabrizio Mulas. 2017. Recommendation in Persuasive eHealth Systems: an Effective Strategy to Spot Users' Losing Motivation to Exercise. In *Proceedings of Second International Workshop on Health Recommender Systems co-located with ACM RecSys 2017, Como, Italy, August 2017 (RecSys'17),* 4 pages.

HealthRecSys'17, August 2017, Como, Italy

1 INTRODUCTION

Persuasive technologies in the eHealth domain are an emerging and promising research field. eHealth persuasive technologies (eHPT) are designed to help people change their habits to overcome their frictions to healthier behaviors [7, 8, 12].

The rise of mobile technologies and the spread of Internet access across the globe favored the development of eHPT. In 2010, we started investigating on the effects of the gamification on users' motivation to exercise [13] and these efforts have led to the development of the u4fit platform[1]. The system connects users with coaches, allowing for a tailored exercise experience at a distance [1].

One important issue that coaches face while following people is that sometimes they stop exercising, without giving any prior notice to them. This aspect is critical, given that a sedentary behavior can negatively affect several social, physical, and mental aspects of the individual, leading, especially on elder people, to chronic diseases.

There are obviously unpredictable cases of users who stop training (e.g., injuries), but there are also behavioral changes in the way users workout that might help predicting such a situation. The automatic detection of motivation decline is still an open issue that, if addressed, will allow coaches (or professionals in general) to act proactively in order to prevent this scenario.

Recommender systems (RS) can help supporting decisions in health environments. As highlighted in [17], when a RS is developed for health professionals (as in our case) they provide information that allows them to address specific cases. Moreover, health RS help providing reliable and trustworthy information to the end users [17]. The goal of health RS is usually to lead to lifestyle changes [16] and to improve the patients' safety [5]. Readers can refer to [3] for a recent survey on health RS.

In this paper, we aim at predicting when a sportsman will stop exercising by analyzing and inferring behavioral patterns from past performances recorded through u4fit. To this end, we train a classifier using as class the fact that the current performance led or not to a subsequent workout within x days (where x is a parameter that sets the granularity with which a coach monitors the sportsmen she follows). Therefore, our objective is to predict if a sportsman will stop exercising within x days. If this happens,

[1] www.u4fit.com. Please note that the coaches marketplace is visible only by setting the Italian language on the platform.

we are able to promptly recommend the sportsman to his coach, in order for her to intervene and prevent him to give up training.

The analysis and the results presented in this paper take into account users' workouts made by means of u4fit for a timespan of approximately two years and a half. The proposed method can greatly help human coaches to intervene in order to reduce the amount of people going to abandon an active lifestyle with the obvious positive outcomes for their health.

The contributions of this paper can be summarized as follows:

- this is the first time in the literature of health RS in which, given the past exercising behavior of a sportsman, he is recommended to his professional (the coach), to trigger her reaction and motivate him not to stop;
- we validated our proposal on a real-world dataset made up of approximately two years and a half of data, by comparing different classifiers on standard accuracy metrics;
- our solution can be embedded in real-world persuasive eHealth systems, thus finding practical and effective applications.

The rest of the paper is structured as follows: Section 2 describes the collected dataset; Section 3 illustrates the features we modeled and the details of the classifiers; Section 4 presents the experimental framework; Section 5 contains conclusions and future work.

2 DATASET

The dataset employed for the analysis consists of a subset of workouts collected by means of the u4fit platform. For each workout we considered the following aggregate statistics:

(1) covered distance (in meters);
(2) workout duration (in seconds);
(3) rest time (in seconds);
(4) average speed (in km/h);
(5) burnt calories;
(6) time elapsed since the previous workout (in hours).

From the total number of available workouts we excluded those whose aggregated statistics are neither related to running nor to the beginner/amateur runner[2]. In particular, we considered only those matching the following constraints: (i) $0.5\,km \leq distance \leq 43\,km$, ($ii$) $workout\ duration \leq 5\ hours$, ($iii$) $rest\ time \leq 1\ hour$, (iv) $average\ speed \leq 16\ km/h$, ($vii$) $burnt\ calories \leq 3000$. The final sample contains 65315 workouts, performed by u4fit users from January 1, 2015 to May 10, 2017.

3 CLASSIFICATION

This section will illustrate the features we modeled and will introduce the classifiers used in this study.

3.1 Features

The values that were measured for each workout have been used to model six sets of features, given as input to the classifiers:

ABS. Each feature contains the absolute value recorded during the workout, presented in the previous list (e.g., the *distance* feature of a workout contains the number of meters covered by the sportsman). In addition to the features of the current workout, we considered also a new feature representing the number of workouts performed by the user previous to the current one, which allows us to contextualize the current performance of the sportsman to his expertise;

INC. Each feature is represented as the increment with respect to the previous workout performed by the user. For example, let $distance_C$ represent the distance covered in the *current* workout and $distance_P$ the one associated to the *previous* workout. The covered distance feature for the INC set is represented as: $(distance_C - distance_P)/distance_P$;

MIN, AVG, MAX. Each feature contains the minimum [average, maximum] of the user's historical training data;

In addition to the features that consider the current workout and perform a matching with the history of a user, we modeled an additional set, named Weekly Load (**WL**), which measures the impact of the previous week's workouts on the current performance. This is achieved by summing the values of the previous seven days for a user, considering the following three features: (i) covered distance, (ii) workout duration, and (iii) burnt calories.

In conclusion, each workout is represented by 34 features.

The class used to train a classifier was binary and it was *1* if the user worked out in the next x days and *0* otherwise.

3.2 Classifiers

In our study, we evaluated and compared the performances of four among the most effective classifiers at the state of the art [6].

Random Forest (RF) is a meta-estimator of the family of the ensemble methods [4]. It fits a number of decision tree classifiers, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all the trees in the forest. See [2] for further details.

Ada Boost (AB) is another ensemble method that, in order to produce the final prediction, combines the predictions from a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) that are fit on repeatedly modified versions of the data [9].

Extra Trees (ET) is also an ensemble method. Similarly to Random Forest, it uses a random subset of candidate features while splitting a tree node; however, instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule [11].

Multi-Layer Perceptrons (MLP) is a neural network that, given a set of features and a target, can learn a non-linear function approximator for classification or regression. It is different from logistic regression, in that between the input and the output layer there can be one or more non-linear layers, called hidden layers [10].

4 EXPERIMENTAL FRAMEWORK

This section will present the experimental setup and strategy, the evaluation metrics, and the obtained results.

4.1 Experimental Setup and Strategy

The experimental framework exploits the Python scikit-learn 0.17.1 library[3] for all the classifiers. The experiments were executed on

[2] We are deliberately not disclosing the total number of workouts before the cleanup given it is a sensitive commercial information.

[3] http://scikit-learn.org/0.17/

an Intel Core i7 2630QM processor equipped with 8GB RAM. Each classification was repeated 10 times with a 10-fold cross-validation.

In order to avoid class imbalance, we tested several undersampling and oversampling strategies on the training set, but the most accurate one was random undersampling. It is worth noting, however, that with $x = 1$, classes are naturally balanced.

We performed three sets of experiments:

(1) **Classifiers comparison.** We compared the classifiers, by running them on all the modeled features, to evaluate the most effective one for $x = 1, 3, 7$.
(2) **Feature sets importance evaluation.** For each set of features, we evaluated its importance by measuring the *feature_importances_* parameter of the RandomForest class in scikit-learn.
(3) **Evaluation of the classifier with less features.** Considering the most performing classifier in the first experiment, we removed one by one the least important set of features. This allows us to evaluate if any of them represents noise in the classification process (i.e., not only a set is less relevant, but it also contains misleading information for our task) or if, even though it is less relevant, each set of features is essential to improve the classification accuracy.

4.2 Metrics

In the evaluation phase, we selected a series of metrics that, going beyond simple accuracy, are suitable to assess the performance of a binary classifier that operates with imbalanced classes (i.e., when one class is much bigger than the other, as in our case).

Accuracy, which measures the fraction of all instances that are correctly classified, can be defined as: $(TP+TN)/(P+N)$. TP are the true positives (i.e., instances of the positive class that are correctly labeled as positive by a classifier), TN are the true negatives (i.e., instances of the negative class that are correctly labeled as negative by a classifier), P are the real positive instances and N are the real negative ones. This metric can be misleading when the two classes are highly imbalanced (a classifier that always predicts the majority class would have a high accuracy). For this reason, we also measured some other metrics that are more reliable in our scenario.

Recall measures a classifier's completeness and is defined as: TP/P.

Precision is a measure of a classifier's exactness and is defined as: $TP/(TP + FP)$.

In our case-study, we considered recall to be more relevant than precision; indeed, motivating a user who does not need it (false positive) is a lesser evil for a trainer, compared to failing to motivate a user who, on the contrary, needs to be motivated (false negative). Considered this, we decided to measure the *F2-score*, which, like other forms of F-measures, is a metric that considers both recall and precision, although weighing the former higher than the latter:

$$F2 = 5 \cdot \frac{Precision \cdot Recall}{4 \cdot Precision + Recall}$$

It is important to notice that neither the recall nor the precision (and, consequently, not even the F2-score) take the true negative rate into account and this is a problem when dealing with highly imbalanced classes [14]; for this reason we measured *Informedness*, defined as: $Recall + true\_negative\_rate - 1$, where $true\_negative\_rate$ is TN/N. It ranges between -1 and 1, where 1 represents a perfect prediction, 0 no better than random prediction, and -1 indicates total disagreement between prediction and observation. As [15] claims, it is the clearest measure of predictive value of a system.

Table 1: Classifiers' comparison for a 1-day prediction, modeling each workout with all the available features.

1 day	RF	AB	ET	MLP
Accuracy	0.7	0.69	0.68	0.58
Recall	0.72	0.74	0.71	0.54
Precision	0.7	0.69	0.68	0.65
F2	0.72	0.73	0.71	0.52
Informedness	0.39	0.38	0.36	0.16

Table 2: Classifiers' comparison for a 3-days prediction, modeling each workout with all the available features.

3 days	RF	AB	ET	MLP
Accuracy	0.69	0.68	0.67	0.62
Recall	0.74	0.74	0.72	0.52
Precision	0.44	0.44	0.43	0.39
F2	0.65	0.65	0.63	0.45
Informedness	0.4	0.4	0.37	0.17

Table 3: Classifiers' comparison for a 7-days prediction, modeling each workout with all the available features.

7 days	RF	AB	ET	MLP
Accuracy	0.69	0.69	0.67	0.53
Recall	0.74	0.73	0.73	0.71
Precision	0.22	0.22	0.22	0.16
F2	0.51	0.5	0.5	0.41
Informedness	0.42	0.41	0.4	0.22

4.3 Experimental Results

4.3.1 Classifiers comparison. Tables 1, 2, and 3 compare the capability of the classifiers to predict if a sportsman will stop working out within the next 1, 3, or 7 days. Figures show that the classifiers are effective for all the metrics we measured. Indeed, for the metrics whose values are between 0 and 1 (i.e., all except Informdness), the values are high; Informedness, whose values are between -1 and 1, returns values close to 0.5, thus in line with the other metrics. This means that, in 70% or more of the cases, we can correctly predict if a sportsman will stop training by monitoring his past performance; therefore, we can assume that the remaining 30% is partially related to unpredictable events for a classifier, such as injuries or personal problems not related to sports. Therefore, by tracking behavioral changes, we can produce effective recommendations for the coaches in most of the cases.

It is also worth noting that the best results are obtained with $x = 1$, i.e., when classes are already balanced. This happens because, when x grows, we remove information with the undersampling, in order to balance the classes. Moreover, sportsmen in our platform tend to work out with a high frequency, so this explains why our

Table 4: Importance of each set of features.

	ABS	INC	WL	AVG	MAX	MIN
***x* = 1**	0.22	0.17	0.17	0.22	0.12	0.1

Table 5: Results returned by training Random Forest with different sets of features (1: ABS+AVG+INC+WL+MAX+MIN, 2: ABS+AVG+INC+WL+MAX, 3: ABS+AVG+INC+WL, 4: ABS+AVG+INC, 5: ABS+AVG, 6: ABS, 7: AVG).

	1	2	3	4	5	6	7
Accuracy	0.7	0.7	0.7	0.7	0.69	0.63	0.65
Recall	0.72	0.73	0.73	0.73	0.72	0.65	0.68
Precision	0.7	0.7	0.7	0.7	0.69	0.64	0.66
F2	0.72	0.72	0.72	0.72	0.71	0.65	0.67
Informedness	0.39	0.39	0.39	0.39	0.38	0.27	0.3

algorithm can track more effectively behavioral changes in a short amount of time. In our application scenario, this means that the coach can effectively monitor sportsmen on a daily basis. Due to space constraints, the results of the next experiments will be reported just for this scenario ($x = 1$), since it is the most effective.

In all the three settings, Random Forest is the classifier that performs best (as underlined in the tables), so it is the one selected for the subsequent experiments. For some metrics, however, Ada Boost achieves similar or (very few times) slightly better results.

4.3.2 Feature sets importance evaluation. Table 4 reports the importance of each set of features. The absolute values of the current workout (ABS) and the average historical values (AVG) clearly have more impact in the classification process, although no set of features is much more important than the others.

4.3.3 Evaluation of the classifier with less features. Thanks to the previous experiment, we can rank the sets of features by importance and remove them one by one, to see how the effectiveness of the Random Forest classifier is actually affected by them. The results reported in Table 5 show that the MAX, MIN, and WL sets do not help improve the effectiveness of the classifier and that the MIN set actually represents a small form of noise (indeed, by removing it, the recall increases of 0.1). However, if we remove the other, more important sets of features, the classifier's effectiveness decreases noticeably.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an effective approach to predict if a sportsman will soon stop training. These predictions can be used to recommend the sportsman to his coach, who can motivate him not to stop.

We modeled 34 features, by considering the performance of a sportsman during a workout and by correlating it with his past performances. The results showed that behavioral changes can be effectively tracked by monitoring the workout performances, and that nor the recent history of the user, nor the best or worst performance, add information that can be exploited by a classifier.

In future work we will integrate our RS in the u4fit platform to actually investigate the impact of the recommendations. Moreover, we will also exploit chats between sportsmen and trainers to try to improve the classification accuracy and to investigate if there exist other markers not considered by the presented study.

ACKNOWLEDGMENTS

The authors would like to thank Alessio Demurtas, Marika Cappai, and Stefano Amico for their contribution in this research work.

This work is partially funded by Regione Sardegna under project NOMAD (Next generation Open Mobile Apps Development), through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2013).

REFERENCES

[1] Ludovico Boratto, Salvatore Carta, Fabrizio Mulas, and Paolo Pilloni. 2017. An e-coaching ecosystem: design and effectiveness analysis of the engagement of remote coaching on athletes. *Personal and Ubiquitous Computing* (2017), 1–16.

[2] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32.

[3] André Calero Valdez, Martina Ziefle, Katrien Verbert, Alexander Felfernig, and Andreas Holzinger. 2016. *Recommender Systems for Health Informatics: State-of-the-Art and Future Perspectives.* Springer International Publishing, Cham, 391–414.

[4] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems (MCS '00).* Springer-Verlag, London, UK, UK, 1–15.

[5] Robert G. Farrell, Catalina M. Danis, Sreeram Ramakrishnan, and Wendy A. Kellogg. 2012. Increasing Patient Safety Using Explanation-driven Personalized Content Recommendation. In *Proceedings of the Workshop on Recommendation Technologies for Lifestyle Change (LIFESTYLE 2012) (CEUR Workshop Proceedings).* CEUR-WS.org, 24–28.

[6] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do We Need Hundreds of Classifiers to Solve Real World Classification Problems? *J. Mach. Learn. Res.* 15, 1 (Jan. 2014), 3133–3181.

[7] Brian J Fogg. 1999. Persuasive technologies. *Commun. ACM* 42, 5 (1999), 27–29.

[8] Brian J Fogg. 2002. Persuasive technology: using computers to change what we think and do. *Ubiquity* 2002, December (2002), 5.

[9] Yoav Freund and Robert E Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* 55, 1 (Aug. 1997), 119–139.

[10] M.W Gardner and S.R Dorling. 1998. Artificial neural networks (the multilayer perceptron): review of applications in the atmospheric sciences. *Atmospheric Environment* 32, 14 (1998), 2627 – 2636.

[11] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely Randomized Trees. *Mach. Learn.* 63, 1 (April 2006), 3–42.

[12] Wijnand IJsselsteijn, Yvonne de Kort, Cees Midden, Berry Eggen, and Elise van den Hoven. 2006. *Persuasive Technology for Human Well-Being: Setting the Scene.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1–5.

[13] Fabrizio Mulas, Paolo Pilloni, Manca Manca, Ludovico Boratto, and Salvatore Carta. 2013. Linking Human-Computer Interaction with the Social Web: A web application to improve motivation in the exercising activity of users. In *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom).* 351–356.

[14] David M.W. Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies* 2, 1 (2011), 37,63.

[15] David M. W. Powers. 2012. The Problem with Kappa. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12).* Association for Computational Linguistics, Stroudsburg, PA, USA, 345–355.

[16] Haggai Roitman, Yossi Messika, Yevgenia Tsimerman, and Yonatan Maman. 2010. Increasing Patient Safety Using Explanation-driven Personalized Content Recommendation. In *Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10).* ACM, New York, NY, USA, 430–434.

[17] Martin Wiesner and Daniel Pfeifer. 2014. Health Recommender Systems: Concepts, Requirements, Technical Basics and Challenges. *International Journal of Environmental Research and Public Health* 11, 3 (Mar 2014), 2580–2607.