# A New Framework for Increasing User Engagement in Mobile Applications Using Machine Learning Techniques

Merve Gençer[1], Gökhan Bilgin[2], Özgür Zan[1], and Tansel Voyvodaoğlu[1]

[1] Done Info. and Com. Systems Istanbul, Turkey
{merve,ozgur,tansel}@donetr.com
[2] Yildiz Technical University, Dept. of Computer Engineering, Istanbul, Turkey
gbilgin@yildiz.edu.tr

**Abstract.** In this paper, it is proposed to build a new framework which anticipates mobile user status and behavior characteristics with the aim of increasing user engagement and provide stickiness in mobile applications (iOS-Android) by using machine learning techniques. Motivation of this study is based on the idea of collecting data from users by non-survey methods because data collection from surveys may mislead the system model according to the literature researches on user experience. User behavior includes forecasting next usage time of the user, user motivation type, user mastery level and current context of the user. In order to find relevant patterns, usage data is obtained from pilot mobile applications at first and then they are processed according to the chosen machine learning algorithm.

**Keywords:** Engagement, stickiness, mobile applications, machine learning, user experience, human computer interaction, context awareness, mobile devices, push notification.

## 1 Introduction

Smart phone adoption has surpassed that of any consumer technology in history. It is 10 times faster than that of the 1980's computer revolution, double times faster than that of 1990's Internet explosion and triple times faster than that of recent social network adoption [1]. Exponential proliferation of mobile applications and user experience provided by these applications caused this high adoption rate. As of June 2012, Apple App Store operates in 155 countries with more than 650.000 applications, and as of March 2012 [2] Google Play operates in 129 countries with more than 500.000 applications. In June 2013, total number of Android applications is expected to exceed 1.000.000 in Google Play. Apple's App Store added about 75.000 apps between Sept. 12, 2012 and Jan. 7, 2013 which constitutes more than 19.000 new apps every month [3].

We assert that in order to have a competitive advantage, a mobile application developer company should target "engagement" and "stickiness" as a key performance indicator because number of downloads alone does not indicate sustainable success. According to the statistics of Flurry who is an iPhone app metrics company, after 30

A. Marcus (Ed.): DUXU/HCII 2013, Part IV, LNCS 8015, pp. 651–659, 2013.

days, a free iPhone app generally loses 95% of players [4]. We define engagement as the frequency of usage per user per month multiplied by average session duration per usage in a month. Similar to the literature, stickiness is accepted as a measure of motivation for long term loyalty [5] and operationalized as the number of daily active users (DAU) divided by the number of monthly active users (MAU).

There are several methods for increasing engagement. For example rewards and material incentives provide an extrinsic motivation for users. Another method for creating re-engagement is sending push notifications to mobile phones, which trigger usage. Although these methods seem to be efficient per se, none of them are able to extract contextual information, personal tendencies and consider timing and personal motivation types. Thus, there is a need for a better method which is able to extract user's context (e.g. is she alone or in a social context? is she at home, at work or at a restaurant? is she mobile or settled? is she sleeping or working? is she casual and available or busy and not available?), personal tendencies (such as timing of sending push notification messages. e.g. is she likely to read (e.g. news) from her tablet at nights while she is taking a rest or is she likely to use the application while she is on the bus?) and personal motivational type based on Bartle's research [6].

In literature, user data collection and building a consistent decision system have been studied by many researchers. The most popular way for collecting data from users is making them answer to proper questions (surveys) in order to catch their life styles, habits, hobbies, to do lists and so on. Moreover, frequent locations of users are important to extract context information. An experimental study by combining locations with proper web services using a decision tree for determining user behavioral patterns is introduced in [7]. On the other hand, using web-based statistics and general information like weather forecast, season, maps, etc. have been used along with the information gathered from user surveys in [8]. There are also studies which concern only in locations of users and how much time they spent in the proper locations in order to decide users' contexts [9]. The survey-based methods and observation inference-based (non-survey methods) have been compared in [10]. Survey-based data collection provides quick building database with less effort as an advantage. Nevertheless, data collection from surveys does not always show the truth. According to Jacob Nielsen [11], it is important to collect data from users by searching how they actually use, not by asking questions and expecting them to give the true answers. Survey-based data collection methods are doubtful for reflecting the truth, which means they might mislead the system [12].

In this study, data collection is achieved by observation inference-based way which monitors users in their daily monotones. Since this method does not involve in any reference information (e.g. surveys), knowledge discovery and information extraction cover wide range of difficulties. After the collection of the user data, formatting and transforming them to the user-features are implemented. A machine learning framework for the estimation of next usage time, context of the user (location and commuting route) and estimating user mastery level have been proposed in a modular structure.

# 2 Methodology

The main target of the proposed system is providing sustainable engagement of users and mobile applications. In order to achieve this goal, it is planned to model an intelligent and dynamic system that recognizes the IDs of the mobile users based on their previously learned routines (patterns). Hence, future behaviors of users are estimated to keep them using the application. By taking into account that each user may have different behavior patterns, the system must produce different outputs so as to be as intelligent as it is expected. Thus, every user is evaluated separately and the system produces different models due to the different usage characteristics of users.

Methodology for the proposed system is summarized and shown in Fig. 1. It consists of data collection step from users, pre-processing step for preventing database from outlier samples, preparation of the data for providing appropriate inputs for the machine learning algorithms and final decision will be given according to the modeled structure.

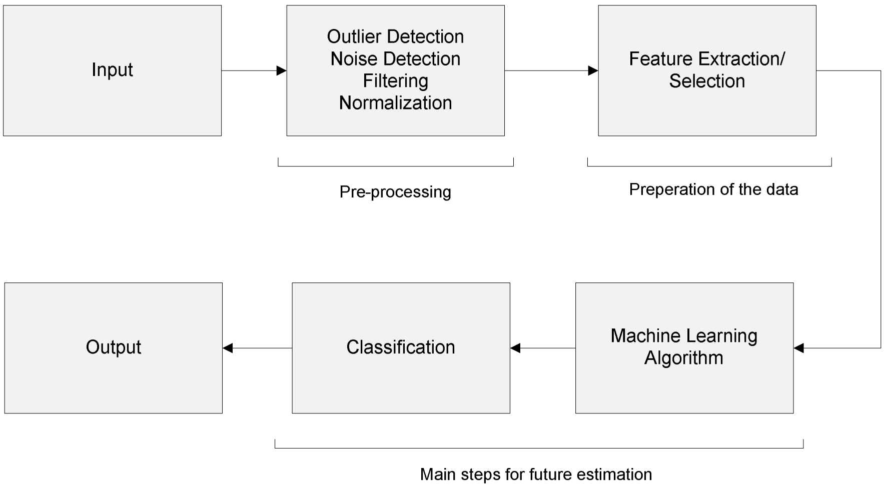


**Fig. 1.** Flow chart of general system framework

In the following sections, methodology for this study is described in details. A general view to the system architecture is visualized and communication between clients and servers is explained in Section 2.1. Data collection step and the specifications of the collected user data are described in Section 2.2. The preparation of the data for machine learning algorithm is recounted and exemplified with real user data in Section 2.3. Since the study is very extensive and needs extra attention, the ongoing studies on machine learning algorithms and future work are stated in Section 3.

## 2.1 System Architecture

The scenario behind proposed system works as in Fig. 2. The Android/iOS mobile devices as the client, which are using the application, send data to “DoneLabs Client

(DC)", the background service for collecting data from users. DC receives data from the client, connects to "DoneLabs Database (DD)" which runs on Amazon Web Services (AWS), and inserts data to the proper tables of DD. Database can be accessible from DC for insertion and "DoneLabs Machine Learning Client (DMLC)" for selection. DMLC is the core client which applies the pre-processing, feature extraction and selection steps and all the calculations for chosen machine learning algorithm. DMLC requests user data from DD in certain periods of time in a day and produces models for each user by using data from DD. DMLC sends the model it produces to "DoneLabs Decision Maker (DDM)" after execution. The model delivered to DDM is stored for model requests from client itself. Models stored in DDM are refreshed by the new ones sent from DMLC every day.

The key point in this model is the separation of DMLC and DDM from each other. While DMLC is producing new outputs (models) for the users as an offline system, DDM only stores the models for a fast response in need of an estimation for a user.

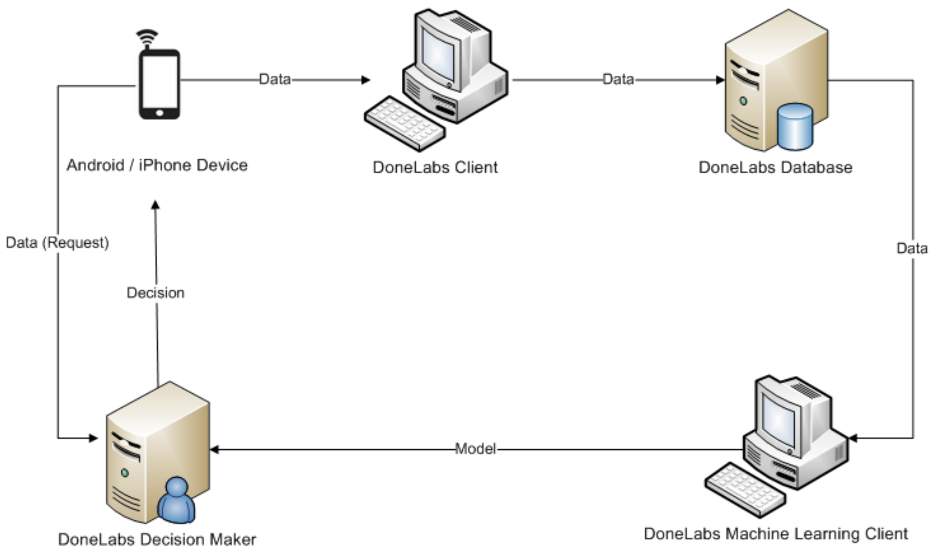


**Fig. 2.** System architecture

## 2.2 Generating the Database

Machine learning algorithms need consistent, featured, assorted types of data in order to work properly and produce reasonable outputs. Thus, data collection step is very important. Decision of the data types acquired from a mobile device has been made experimentally. A user tracking library has been developed and imported to the popular DoneLabs mobile applications to collect data from the users. There are three types of data collected for this study:

- User static information

- User action information
- User dynamic information

**User Static Information.** It simply covers the general information about the device and the operating system of the user. Device name and model, operating system name and version, application version, user tracking library version are gathered and sent to the database.

**User Action Information.** It covers the most important features of the database. All the actions of the user in the application are tracked and recorded to the database.

User actions include date and time information of the following actions:

- Openings and closings of the application,
- Entering background and foreground,
- Clicked buttons,
- Opened pages

Names of the buttons and pages of interest are also recorded. User location is also tracked as a user action for every 50 meters change.

**User Dynamic Information.** It covers the information about the device, that can be changed according to the environment. Battery status (plugged, unplugged, charging, discharging, not-charging), battery level, brightness level, headphone status (plugged, unplugged), internet connection status (Wi-Fi or Cellular), volume level of the application and gyroscope information are gathered as dynamic information from the users. Dynamic information is useful for estimating the context information of the user.

Furthermore, Gyroscope information gives the orientation of the mobile device in space. It consists of 3 dimensions, pitch, yaw and roll. The values of yaw, pitch, and roll changes due to the rotations around x, y and z axes of the device. By knowing the orientation of the device, it is helpful to figure out the context of the user. If no great value changes occur on gyroscope values, it can be considered as the device is fixed to a position. On the other hand, if the gyroscope values change in great differences, the user can be considered as holding the device in hand. Combining gyroscope info with location and sensor information can provide more accurate user context estimations.

Since this study aims gathering data from users without any reference (e.g. surveys), it is important to track users in different aspects of data. Action information is the key point where the users can be tracked as the actions they take in an application, so it may provide tracking the user's motivation, mastery level, spending habits, frequency of usage etc. This information is considerably valuable for estimation of next usage time and user's motivation to quit using the application. Static information may provide user device choice trends, updating trends and her desire for new updates.

In order to ensure the privacy of the mobile user, all the data is collected anonymously (user profile information is neither requested from the user, nor sent to the data-base) and users are given User ID's.

### 2.3 Formatting the User Info

The period of data collection has been considered as three months which is expected to be sufficient in order to satisfy consistency between research time and cost constraints. Within this period, over 1.710.882 action info, 4.558 static info, 37.578 sensor info and 76.095 gyroscope info for 2.883 users have been recorded.

The first goal is to estimate the next usage time for a user, so the dates and times for application-open, application-close, enter-background, enter-foreground actions are prioritized. The actions, application-open and enter-foreground, are grouped as "start of a session" and application-close and enter-background are grouped as "end of a session". The dates and times for these groups of actions have been gathered from the database for each user as well. The sessions for each user have been stored per day as matrices of vectors. Using these stored session and daily usage matrices, weekly and monthly usage times and durations for users have been extracted. This extraction shows usage patterns of the users for the application of interest.

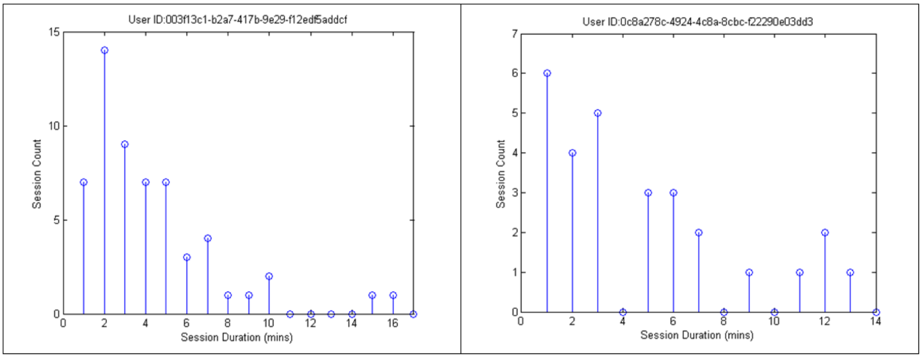


**Fig. 3.** Session durations and counts of two different users in one month

In Fig. 3, the session durations are shown for two different users in horizontal axis. The usage counts of those session durations are also shown in vertical axis. From that information, total usage times and average usage times of the users can be gathered for one month. Other features gathered from session information are shown in Table 1. Starting and ending hours of those sessions have been stored in another structure for future work.

In data acquirement step, some outliers may exist which are irrelevant with characteristics of users. For example, (a) if the user opens and closes the application for a short time, (b) if the client cannot send data to the server before the user closes the application, or data is lost on the way, (c) if there is a conflict between user action data, e.g. application close or enter background actions never arrived before starting a new session, (d) if sensor information of a user arrives all 'unknown' for a session may cause outliers.

Outliers are ignored before feature extraction step. Other than that, there are also noises in daily usage and monthly session user histograms. To eliminate noisy values of users, a data smoothing filter is applied to the histogram values. In this way average usage time and proposed features give more accurate results.

**Table 1.** Usage data of random 10 users of the application "Mobil Lig" (iOS)

| UserId | TotalUsageTime/Month (min) | MaxUsageTime/Month (min) | AverageUsageTime/Month (min) | Total Usage Count/Month |
|---|---|---|---|---|
| 003f13c1-b2a7-417b-9e29-f12edf5addcf | 239 | 14 | 2.779069767 | 12 |
| 0814f8c4-328a-4d9f-b6ee-63905d244db4 | 120 | 7 | 1.935483871 | 9 |
| 0c8a278c-4924-4c8a-8cbc-f22290e03dd3 | 133 | 6 | 1.927536232 | 10 |
| 0f66b7e8-3d1e-4b8f-a50b-11e5929b3b48 | 225 | 8 | 1.7578125 | 15 |
| 10206e4e-1aff-46da-b17a-10b2bbba0ece | 110 | 9 | 1.774193548 | 10 |
| 128b097c-f29e-477b-958e-a9d66123bcb3 | 127 | 5 | 2.351851852 | 9 |
| 16dc422c-10b1-450e-a396-f68c727d4b59 | 200 | 6 | 1.709401709 | 14 |
| 1f16e1f9-25d2-4ab7-b0d1-52e84af0397d | 91 | 5 | 1.895833333 | 8 |
| 20f092f5-7f1e-4fa0-8a4d-cd7486d910af | 209 | 4 | 1.6328125 | 14 |
| 22b5ec9f-e352-4bfc-9d03-56f495a319c1 | 810 | 17 | 2.612903226 | 23 |

Briefly, in the feature extraction algorithms data are transformed to get the most informative features, such as in principal component analysis (PCA) and linear discriminant analysis (LDA) etc. [13]. On the other hand, feature selection algorithms seek for discriminative features without transformation using some criteria, such as information gain, correlation etc. [14]. Feature extraction is still an ongoing subject of interest of this study. Evaluations for more reasonable features that reflect the user usage characteristics are continuing.

## 3 Conclusions and Future Work

Collection of user data and feature extraction/selection step for machine learning algorithms cover the most of this study to provide accurate results. After completing mentioned steps above, data is ready to feed the input of the machine learning algorithms. In order to get realistic results from chosen machine learning algorithm, inputs have to be prepared regularly. The decision of which machine learning technique is the most appropriate will be made after testing various techniques.

The location information of the users is important for estimating the contexts of them. The user might be in the class, at home, at work, in a concert, in a coffee shop and so on. The user location is tracked in existing user tracking libraries. When it reaches the optimum amount of data, feature extraction/selection can be implemented and data can be given as input to the chosen machine learning algorithm in order to estimate future or current context of the user. The main purpose for doing the estimation on context information is predicting the true place/environment as well as the time for customizing and sending push notifications.

Future works of this study is intended to fulfill the following objectives respectively:

- Next usage time of the user can be predicted and appropriate push notifications can be sent in order to keep the user using the application

- The optimum days and hours for a user to use the application can be estimated and the best time for sending push notifications can be calculated
- The context of the user can be estimated and push notifications can be arranged and send according to the context of the user
- The users who have a tendency to quit using the application can be detected
- The motivation of a user while using the application can be known (achiever, explorer, socializer, killer according to Bartle Test [6])
- The mastery level of the user can be estimated
- The payment trend of the user can be defined, so the user can be encouraged with rewards and new items

In data modeling phase several unsupervised and supervised learning methods are evaluated and compared by means of prediction/classification accuracy and computational complexity. In this step data are divided into two parts as training and testing sets in order to evaluate general performance of the proposed system. Several machine learning approaches such as k-means, fuzzy c-means (FCM), Bayes classifiers, k-nearest neighbor (KNN), artificial neural networks (ANNs) and kernel based methods such as Kernel Fisher Discriminant Analysis (KFDA) and Support Vector Machines (SVMs) [15] will be studied to enable easier and accurate analysis of our data.

As a conclusion, based on the future calculations and findings, context and tendency of the user and user's main drive of using the application will be predicted in order to increase engagement. For example, the most probable next usage date and time of the application will be forecasted as well as churn probability and date. Actions based on these forecasts include decision of when to send push notifications, customizing the push notification according to user motivation type and considering contextual information.

## References


1. Flurry Research: `http://blog.flurry.com/bid/88867/iOS-and-Android-Adoption-Explodes-Internationally`
2. Apple: `http://events.apple.com.edgesuite.net/126pihbedvcoihbefvbhjkbvsefbg/event/index.html`.
3. Readwrite Mobile: `http://readwrite.com/2013/01/07/apple-app-store-growing-by`
4. Zichermann, G., Cunningham, C.: Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps., Ch. 4. O'Reilly Media (2011)
5. Racherla, P., Furner, C., Babbry, J.: Conceptualizing the Implications of Mobile App Usage and Stickiness: A Research Agenda (2012), `http://dx.doi.org/10.2139/ssrn.2187056`
6. Bartle, R.: Hearts, Clubs, Diamonds, Spades: Players Who suit MUDs, Journal of MUD Research (1996), `http://www.mud.co.uk/richard/hcds.htm`
7. Lee, S.-C., Lee, E., Choi, W., Kim, U.M.: Extracting Temporal Behavior Patterns of Mobile User. In: Fourth International Conference on Networked Computing and Advanced Information Management, pp. 455–462 (2008)

8. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context-aware places of interest recommendations for mobile users. In: Marcus, A. (ed.) HCII 2011 and DUXU 2011, Part I. LNCS, vol. 6769, pp. 531–540. Springer, Heidelberg (2011)
9. Jeong, Y.-S., Oh, K.J., Kim, S.-S., Choi, H.-J.: Context Awareness of Social Group by Topic Mining on Visiting Logs of Mobile Users in Two Dimensions. In: IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, pp. 194–197 (2011)
10. Seo, J., Lee, S., Lee, G.: An Experience Sampling System for Context-Aware Mobile Application Development. In: Marcus, A. (ed.) HCII 2011 and DUXU 2011, Part I. LNCS, vol. 6769, pp. 648–657. Springer, Heidelberg (2011)
11. Nielsen Norman Group, Evidence-Based User Experience Research, Training, and Consulting, `http://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users`
12. Nielsen, J., Budiu, R.: Mobile Usability, 1st edn. The Nielsen Norman Group, Berkeley (2012)
13. Polikar, R.: Pattern Recognition. In: Akay, M. (ed.) Wiley Encyclopedia of Biomedical Engineering. Wiley, New York (2006)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2001)
15. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2007)