Modelling and Predicting User Engagement in Mobile Applications

Eduardo Barbaro [a,b], Eoin Martino Grua [c], Ivano Malavolta [c], Mirjana Stercevic [a], Esther Weusthof [a], and Jeroen van den Hoven [a]

[a] *Mobiquity Inc, Global Analytics Group, The Netherlands*
[b] *IBM, Cognitive and Analytics Benelux - Global Business Services, The Netherlands*
[c] *Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands*
E-mails: eduardo.barbaro@ibm.com, e.m.grua@vu.nl, i.malavolta@vu.nl

Abstract. The mobile ecosystem is dramatically growing towards an unprecedented scale, with an extremely crowded market and fierce competition among app developers. Today, keeping users engaged with a mobile app is key for its success since users can remain active consumers of services and/or producers of new contents. However, users may abandon a mobile app at any time due to various reasons, *e.g.,* the success of competing apps, decrease of interest in the provided services, etc. In this context, predicting when a user may get disengaged from an app is an invaluable resource for developers, creating the opportunity to apply intervention strategies aiming at recovering from disengagement (*e.g.,* sending push notifications with new contents).

In this study, we aim at providing evidence that predicting *when* mobile app users get disengaged is possible with a good level of accuracy. Specifically, we propose, apply, and evaluate a framework to model and predict User Engagement (UE) in mobile applications via different numerical models. The proposed framework is composed of an optimized agglomerative hierarchical clustering model coupled to (i) a Cox proportional hazards, (ii) a negative binomial, (iii) a random forest, and (iv) a boosted-tree model.

The proposed framework is empirically validated by means of a year-long observational dataset collected from a real deployment of a waste recycling app. Our results show that *in this context* the optimized clustering model classifies users adequately and improves UE predictability for all numerical models. Also, the highest levels of prediction accuracy and robustness are obtained by applying either the random forest classifier or the boosted-tree algorithm.

Keywords: User Engagement, Mobile Apps, Numerical Modelling, Clustering

1. Introduction

Mobile applications (hereinafter “apps”) dominate the digital world today, reaching incredible numbers and showing no signs of slowing down its market growth anytime soon [1]. For example, as of March 2018, there are more than 3.3 million Android applications available [2], with more than one thousand apps being published *everyday* [1]. Mobile apps are not only being published in large numbers, but are also being consumed by users in large numbers, with more than 1.5 billion downloads from Google Play Store every month [3]. A medium of such a large scale leads to a crowded market with strong competition. Under this perspective, *mobile app developers must keep their users active over a sufficiently long period of time to be considered successful*. Recognizing and understanding user motivations are key to leading to a greater app usage [4]. To date, despite significant efforts, over 95% of smartphone owners stop using an app by the end of the third month of download [5]. In other words, the majority of mobile solutions fail to achieve long-term usage. This can be explained by a variety of reasons, such as lack of personalization, user context, and finally failure to seamlessly integrate with other apps or technologies [6–8].

A high disengagement rate is obviously non desirable to app developers, whose success depends on the usage of their app. Furthermore, it is also a problem for researchers and other professionals who use apps to provide services aimed at improving the user's quality of life. Let's look at the e-Health domain as an example. Within e-Health, apps are used as a tool to help users overcome their illness (physical and/or mental) and improve their quality of life. However, for the app to succeed, it must be regularly utilised by the user. Hence, as a crucial quality, it must be engaging.

Crafting personal "smart interactions" is an effective way to ensure that users remain active, on-line, and motivated [9]. Furthermore, tailored interactions aim to maintain, encourage and ultimately increase app usage over time. Take people tracking as an example: mobile location tracking has to be used on a opt-in basis, due to privacy issues [10]. However, once a device is being tracked, apps may send out alerts when the tracking is turned off aiming to prevent the user to go off-line. The nature of these interactions may vary wildly, since it is likely that users react very differently to such interventions [11].

In the context of this study, UE can be intuitively defined as the assessment of the response of the user to some type of activity or service provided by the mobile app. For example, in social networking apps (*e.g.,* Facebook or Twitter) UE is about user's posts, comments, and interaction with other users; differently, in shopping apps (*e.g.,* Amazon or Wish) UE is about the products being purchased, being listed, saved for later purchases, and so on.

Despite there being a good understanding of what is UE in different domains and which factors contribute to it, there seems to be a lack of literature on whether it is possible to predict UE in mobile apps and how different methods perform.

In this study, we provide evidence that **it is possible to predict the engagement of mobile app users with good levels of accuracy**. We achieve this result by characterizing and evaluating a framework for predicting user engagement of mobile apps. The framework is based on the application of different types of numerical models, *i.e.,* survival, counts, and classification. The numerical models take as input a minimal set of information about the user, which are relatively straightforward to collect at run-time, *e.g.,* the current point balance of the user (assuming the app is employing a potentially implicit gamification mechanism), the time of the last interaction with the app, geographic position, etc. In this study, we explore four different types of numerical models, namely: (i) survival analysis, (ii) negative binomial regression, (iii) random forest, and (iv) gradient-boosted trees. In order to complete our approach, one of the most important steps to achieve better predictions is to group users based on their past behaviour [12]. In that way, it is possible to separate - or "cluster" - users based on how (often) they interact with the mobile app. Therefore, we also incorporate a clustering algorithm to our proposed framework, aiming at targeting user interactions more accurately by means of drawing similarities between users [12].

We empirically evaluate the performance of our proposed numerical framework in predicting UE on an industrial dataset, which has been built in the context of a *real mobile app* in the area of waste recycling. The dataset is composed of approximately 27,000 entries distributed over 1,500 unique users.

Summarizing, the main contributions of this study are:

- a reusable framework for modeling and predicting UE in mobile apps;
- a characterization of UE by means of 4 different types of numerical models;
- the empirical evaluation of the prediction accuracy of the 4 different types of numerical models in the context of a waste recycling mobile app.

The contributions above benefit both mobile apps developers and researchers. Developers can re-use the proposed framework for accurately predicting the engagement of their users at run-time and counteract it in a timely fashion (*e.g.,* by sending a push notification for triggering new conversions) - see [13], and (ii) learn from the evaluated numerical models which one is better suited for their own mobile app. We support researchers since we (i) provide evidence about how various numerical models can accurately estimate UE in mobile apps and (ii) provide a framework for modeling and predicting UE, which can be further extended or used in other scientific studies.

It is important to note that the the aim of this study is not to provide a general solution for predicting UE for all mobile apps, instead we aim at providing (i) evidence that it is possible to predict UE with good levels of accuracy and (ii) a flexible framework for modeling and predicting UE in mobile apps which can be re-used by both researchers and practitioners in other projects, provided that it will be customized according to the app under consideration, its usage scenarios, and the available data.

The remainder of this paper is organized as follows. Section 2 presents the fundamental background needed throughout this research. Section 3 presents the modeling framework, whereas the results of the evaluation of the prediction accuracy of the modeling framework are reported in Section 4. Finally, Section 5 discusses and puts into context the obtained results and Section 6 closes the paper.

2. Background

In this section we provide background information about the definition of user engagement in the context of mobile apps (Section 2.1) and present the waste recycling app dataset (Section 2.2).

2.1. Defining User Engagement

User engagement is not a trivial concept to define, especially in the mobile segment. As a first attempt, UE can be described as a proxy for quantifying an outcome or, more generically, interpreting an action. In [14] the authors summarized and combined several prior definitions of engagement. They argue that UE consists of users' activities and mental models, manifested as attention, curiosity and motivation. As shown in Figure 1, UE can be seen as a process composed of four main steps, namely: users (i) start engaging with a mobile application, (ii) remain engaged, (iii) disengage, and finally (iv) potentially re-engage. Building on that argument, in a later study, the same authors argued that engagement is not only a product of experience, but also a cycle-process that depends on the interaction with technology [15]. Closely related to [15] and [16], [17] defined UE as the quality of the experiences that emphasize the positive aspects of the user interactions.

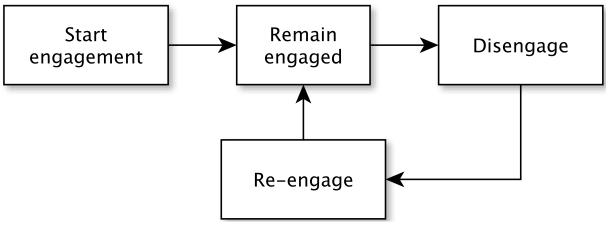

Figure 1. Overview of UE life cycle. The arrows indicate the possible places of interaction with technology. Figure inspired in the four-step engagement process proposed by [14] and [15]

More recently, on-line behaviour was analysed to better understand the temporal evolution of UE in massive open on-line courses [18] . Their findings suggest the use of diverse features - such as last lecture watched, last quiz taken, and current/total number of posts - as good quantitative indicators for modelling UE at different points in time. In their study, they used these parameters to accurately predict student survival rates already at the beginning of the course [18]. Another paper modelled UE for on-line health communities [19]. They provide a variety of predictive features, e.g. total number of posts and threats initiated, to a complex set of classification models, amongst them random forest and survival models. They claim to have found that UE in on-line health communities can be achieved by seeking emotional support and by being actively involved in companionship.

In the remaining of this section we introduce the fundamental concepts associated with the numerical tools used to model UE in mobile apps. Naturally, the first point to address here is to properly *classify* if a customer is engaged or not at the present time. Different definitions can be used - or combined - to address that. Here, we discuss:

- The application is still installed on their phone after a certain number of days;
- The number of user activities is bigger than a given threshold;
- The frequency of user activities is higher than a given threshold.

One of the simplest definitions available is called **User Engagement Index** (UE$_I$). The UE$_I$ compares the time of inactivity with the time the customer has been engaged. Mathematically it reads:

$$UE_I = \frac{LastEvent - FirstEvent}{Today - FirstEvent}, \tag{1}$$

where all the terms on the right-hand side are dates. We see in Equation 1 the ratio of the time difference between both last event and present time to the time of the first interaction. If UE$_I$ > 0.5 (where 0.5 is a threshold defined a priori) the user is considered engaged *today*.

Another possible way to determine UE is by defining a threshold on the **recency** (R). This threshold has to be calculated to determine if the time between actions is (long)short enough for the user to be considered (dis)engaged. Recency is trivially defined in Equation 2:

$$R = \Delta t, \tag{2}$$

where Δt is the time past between one action and its *subsequent* action. In doing so, user engagement based on recency (UE$_R$) can be calculated for every interaction, and not only for the last one as in UE$_I$.

We base the choice of threshold to determine UE$_R$ on the statistical distribution of R. The threshold is set as being at the edge of one standard deviation from the average recency. By doing so, we ensure that to be considered disengaged the user's recency has to be less than around 32% of our entire sample recency. That is a compromise between allowing for later re-engagement (by not tackling only users at the very end of the distribution, i.e. almost totally disengaged) and not sending too re-engagement messages to still engaged users (users close to the center of the distribution). Similarly to UE$_R$, we explore the fact that user engagement can also be defined by setting a threshold on the **total number of actions** (A$_T$) a user performed within a given time frame. Mathematically, it reads:

$$A_T = \sum_{t_0}^{t_N} A(t), \tag{3}$$

where t_0 and t_N are respectively the initial and final times of the counting. Every user surpassing a given threshold can be considered engaged.

2.2. The Waste Recycling App Dataset

In this study, we use a dataset from a mobile app that promotes waste recycling. The app grants points every time an event is performed by the user, *e.g.,* disposing trash in their selected bins, reading educational material, or inviting friends to join the app. These points can then be redeemed for rewards at selected partners, such as savings on local shops or discounts on sustainable goods. Extending the framework described in [20] for tablets, we argue that the app needs to be designed and optimized having in mind that the user is most likely on their mobile phone either redeeming points at a shop or collecting points at the recycle bin. That is fundamental to create an intuitive interface that facilitates these activities and promotes engagement.

The dataset contains approximately 27,000 entries distributed over 1500 unique users and 122 variables. The data was collected between April 2015 and January 2016. Each entry of the dataset contains the following 6 features:

(1) the current point balance of the user,
(2) the time of the user's last event within the app,
(3) the number of days since the last event,
(4) the current weekday,
(5) the current ZIP code,
(6) the current geographical position of the user in terms of latitude and longitude.

We expand each of the 27,000 entries of the dataset to contain 122 unique variables in total. We achieve that by first generating combinations of these variables, e.g. *number of days since the first event during weekdays* or *time of the user's last event within the app during a weekday/weekend*. We then proceed to calculate the following statistics (*max/min/mean/med/sum/sd*) for all of the variables. That allows for more feature creation, e.g. *standard deviation of the number of days since the first event during weekdays*. We calculate the most simple statistics such as *mean of the current point balance* or *minimum number of days since last event*, but also combinations of variables with statistics - such as *median of the minutes since last event per user in a certain zip code*, or the *standard deviation of the number of days since the first event during weekdays*. Note that geographical position provides more detailed information than just zip-code, given that there may be more than one recycle bin in a given area.

Figure 2 shows the strategy that we follow for splitting the dataset into four main subsets, namely: training, test, cross-validation, and validation sets [21].

Specifically, a fraction of the dataset (60%) is used to train our models and the remaining data to test (20%) and cross-validate (20%) their performance. The last three parts (observation 1,2,3) are the validation sets. They also start at the beginning of the dataset (April, 24) and continue after the end of the training period - as shown in Figure 2[1]. It is important to mention that the validation sets only contain users that remained active, or started new interactions after the training period. Those are depicted in red in Figure 2. We highlight that this setup is general/flexible enough to be used by all our numerical models.

[1] For simplicity, we extrapolate the use of the term training period to indicate the period between April 24 and Dec 1, 2015.

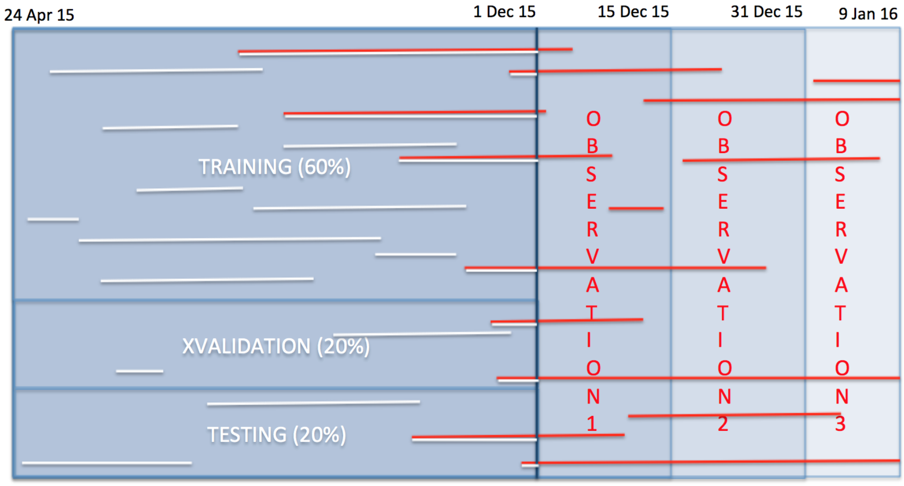

Figure 2. Sketch of users lifespan over time. The red lines indicate customers engaged after the end of the training period.

Concerning the definitions of UE, in this study, we rely on the definitions based on *recency* (see Equation 2) and *total actions* (see Equation 3). The user engagement index (see Equation 1) does not fit the purpose of this study since it is a too coarse-grained definition and it does not provide any information concerning the daily evolution of UE. In our case, the threshold for recency is set constant and equal to 9 days. For the counting model, we choose a threshold of 5 interactions per 2 weeks. These thresholds have been defined based on (i) a number of informal interviews we had with professionals working in the company developing the waste recycling app and (ii) the need to simulate the quick reaction of the app as soon as the users start to be disengaged. We *extensively experimented with a series of other levels of the recency and interaction thresholds around the ones used in this study, and the results of the re-applied models did not significantly vary in all the cases* ($< 5\%$). For the sake of brevity, we do not report the whole set of the performed replications in this study. Finally, it is important to note that the values of the thresholds used in this study strongly depend on the application domain (*i.e.,* waste recycling, in our case); we suggest researchers and developers willing to re-use our framework in other domains/organizations to fine tune the selected thresholds according to the specific characteristics of the app under consideration and its typical usage scenarios (*e.g.,* social media users may be considered disengaged much earlier than after 9 days of total inactivity). In addition to that, note the modelling results - especially the quantitative component - discussed here remain specific for this dataset. Hence, it should not be directly transferred to other application domains. Instead, the main contribution of this paper lies on the fact that we show, by means of different types of algorithms, that it is possible to accurately predict user engagement as well as a reusable framework that can be used to better understand UE in mobile apps.

3. Modelling User Engagement of Mobile Apps

In this Section, we detail our modelling strategy and explain the multiple steps and assumptions we make to predict UE or counts (actions) until disengagement. Here, despite the numerical model we choose, the first step is to describe the process of assigning our users to different groups, the so-called *clustering process*. In doing so, we are firstly grouping similar users together in order to reduce uncertainties and improve the predictability of our numerical models [12].

3.1. The clustering model

In this study, we use a modified Agglomerative Hierarchical Clustering (AHC) model [22]. That means, we assign each data point to one exclusive cluster, and then combine the two clusters that are closest to each other. This process is repeated until there is only one cluster left - containing all the observations. We utilize average linkage to perform the clustering, *i.e.,* the average distance between each point in one cluster to every point in the other cluster. We use the so-called Pearson-γ correlation as our criterion to select an appropriate number of clusters [23, 24]. This metric looks at the correlation of all the distances between data points and a binary matrix, that is equal to zero for every pair of observations in the same cluster and equal to 1 in case points are in different clusters.

Hierarchical clustering methods require a distance metric to define similarity between two observations. Here, we implement the so-called Gower's metric [25] with optimal weights, as proposed in [26]. This metric allows for the calculation of the dissimilarity between rows of our dataset for nominal, binary, and ordinal variables. The optimization is done with the intent to maximize the cophenetic correlation coefficient (CPCC), see [27]. The CPCC is the correlation between the distance matrix used for the clustering and the cophenetic distance matrix of the resulting hierarchical clustering. This cophenetic distance matrix is calculated as the distance at which two observations are combined into one cluster.

The optimization of the CPCC is done through the use of the L-BFGS-B method (Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with Bounds), a Quasi-Newton algorithm which uses the first order derivative of a given function and an approximation of its second-order derivative to obtain the extrema of the given function (non-linear optimization) - see [28]. We apply this method to iteratively search for an optimal set of weights to Gower's metric to optimize the CPCC of the resulting agglomerative hierarchical clustering. The bounds of the L-BGFS-B method are set to [0,1] to ensure no weight is negative. Next to that, we also use an approximation of the analytical derivative of the CPCC with respect to the weights to ensure we do not have to use finite differences for the L-BFGS-B method, hence significantly reducing computation time [26].

As the last part for the configuration of our clustering, we choose which variables we consider to be used for clustering. The variables we pick determine what our clusters represent. As an initial set of variables for our clustering algorithm, we choose all 122 variables mentioned above. In this context, our clusters represent different characteristics of the users and their behaviour, ranging from regional data to frequency of use and point collection. Users in the same cluster are thus expected to be more similar when it comes to app behaviour and geographical location compared to those in other clusters. Hence, these clusters capture useful information for our different user engagement models to use in their predictions.

With our users set to a particular cluster, we use these results as a predictor of UE improving modelling results [26]. In the next subsections, we explain the numerical models we use to predict UE for every user. We detail the three different model types - *survival, counts, and classification* - to evaluate the potential of each approach and the validity of their assumptions.

3.2. The Cox proportional hazards model

In this subsection, we explore the Cox Proportional Hazards model [29]. The Cox Proportional Hazards (CPH) model is a very popular regression model that calculates survival times based on the effect of selected predictors. It becomes especially useful here since our predictors are (highly) non-linearly related and we may not know their distributions beforehand. Another advantage of the CPH model is the fact that it is able to handle missing observations, i.e. sparse user interactions. The CPH model only requires as independent parameters (i) the time of the analysis and (ii) the engagement status. In our case, the status indicates if disengagement happened or not at any particular time. With these two parameters, we estimate two functions called conditional survival and baseline hazard. The former provides the probability of not experiencing disengagement while the latter gives the probability that disengagement will occur up to a given time - see [30]. In our context, the term *proportional hazard* indicates that the hazard ratio comparing two observations is constant in between events. Furthermore, the impact of the different factors on the hazard remains constant over time [31]. We use a threshold equal to 0.5 to determine engagement/disengagement and follow [32] to ensure monotonic Receiver Operating Characteristic (ROC) curves by means of the nearest neighbor method.

3.3. The negative binomial model

Here, we describe a regression model with count data (negative binomial model). This approach is interesting because, in contrast to the CPH model, it allows us to model re-engagement. Here, rather than the time of disengagement, we aim to predict the total number of actions before disengagement. The idea is to target smart interactions aiming to keep the user engaged if the actual counts fall too close from the prediction of disengagement. Briefly, the negative binomial (NB) distribution is the distribution of the number of trials (actions) needed to get a fixed number of failures (in our case disengagement) - see [33]. This distribution describes the probabilities of the occurrence of integers greater than or equal to 0. By analyzing the distribution function, we can set a threshold on the probability of disengagement and extract the number of counts before disengagement. NB is specially suitable to model over-dispersed count variables. This specific regression method is implemented by fitting a generalized linear model using a boosting algorithm based on component-wise univariate linear models - see [34, 35], and [36]. In each boosting iteration, a simple linear model is fitted (without intercept) to the negative gradient vector and in the update step only the best-fitting linear model is used. This machine learning method optimizes prediction accuracy and carries out variable selection. In our case, we perform 500 non-centered boosting iterations with a step length equal to 0.05.

3.4. The random forest model

The RF model [37] basically creates many random independent subsets of the dataset containing features and a training class. In our case, the features are the information about the user, e.g. number of interactions and type of interaction, and the class is simply a flag indicating engaged or disengaged at that particular moment. These subsets are used to create a ranking of classifiers. It is important to state that RF models are typically accurate and computationally efficient. The randomness component ensures the RF model to generalize well, and to be less likely to overfit [38].

In contrast to the other approaches, the RF model is not predicting days (CPH) or counting actions to disengagement (NB). Here, based on past behaviour, we use the RF algorithm as a classifier (engaged/disengaged) *at the moment*. That means, we obtain as outcome a probability value ranging between 0 and 1. With that in hand, we define a cutoff threshold to determine if the user is engaged or disengaged. For our dataset, the cutoff threshold is chosen as equal to 0.42 as it maximizes the F1 score [39].

Interestingly, the RF classification has a predictive component. This is because the RF model simulates UE$_R$. As shown in Section 2.2, this metric is defined as the difference in days between an action now and in the next 9 days. Due to that, we assume that our results are "valid" not only *at the moment* but within the recency threshold as well. Note that this links the validity of the RF model to the recency threshold. This further motivates the choice of a short recency time, just enough to allow the app developer to send re-engagement notifications and monitor their effectiveness.

To build this random forest model, we use 1000 non-stratified trees with replacement (to decrease variance without increasing bias). The number of variables randomly sampled as candidates at each split equal to 10. We use a 10-fold cross validation with 5 repeats to augment model accuracy without increasing bias. The cross validation involves splitting our dataset into 10 subsets. Each subset is then put apart and the model is trained on the leftover subsets. The overall accuracy of our model is then determined after averaging the results obtained with the 5 individual repeats.

3.5. The XGBoost Model

The last approach used to predict UE takes advantage of boosted-trees algorithms. XGBoost is a very popular and scalable end-to-end tree-boosting system [40] currently applied to several different fields of knowledge, such as Physics, stock market prediction, biology and language networks, among others [41–44]. In a nutshell, this classifier constructs trees to make the predictions, but unlike RF, where every tree provides a definite answer and the final result is obtained by a voting process (*i.e.,* bagging), every tree in XGBoost contains a continuous score, which are combined to provide an answer (*i.e.,* boosting). Despite differences with the RF algorithm, the implementation and the use of XGBoost, however, is done very similarly. We utilize the same features to train the model and the output is also a probability percentage indicating whether the user is disengaged *at the moment*. We use a small learning rate equal to 0.001 to ensure convergence and error minimization. The maximum depth of each tree is capped at 15 and the maximum number of trees is fixed at 1000 (similar to RF).

4. Evaluation of Predicting User Engagement of Mobile Apps

In this section we report on the empirical evaluation of the proposed modeling framework *in the context of a waste recycling mobile app*. Specifically, we aim at answering the following research questions:

- RQ_1 – To what extent using a clustering algorithm impacts the accuracy of UE prediction?
- RQ_2 – Which types of numerical models provide the most accurate UE prediction?
    - ∗ $RQ_{2.1}$ – What is the prediction accuracy of the Cox proportional hazards model?
    - ∗ $RQ_{2.2}$ – What is the prediction accuracy of the negative binomial model?
    - ∗ $RQ_{2.3}$ – What is the prediction accuracy of the random forest model?
    - ∗ $RQ_{2.4}$ – What is the prediction accuracy of the XGBoost model?

We begin by showing the performance of our AHC algorithm followed by the predictions of UE for our other numerical models. We highlight that a direct comparison between numerical models is not always possible due to their different natures - classification and regression. Thus, we aim to characterize and evaluate them mostly individually. When possible, we try to place our results in a broader perspective. To keep to the brief character of this manuscript we summarize our model results in terms of ROC curves [45]. These are plots that illustrate the performance of a binary classifier, outlining their overall performance. The true positives are defined as the engaged users who were correctly classified as engaged by our model. False negatives represent the engaged users incorrectly classified as disengaged. The area under the ROC curve (AUC) represents the model accuracy, where unity means a perfect model and 0.5 indicates a random result. We use the ROC curve as our performance indicator - similarly to [16] - because it evaluates the performance of the models across all possible thresholds. In addition, AUC delivers a result comparable across all our model approaches and is threshold independent. This is important in our case since the impact of a false positive vs false negative is comparable.

4.1. *Impact of the clustering model (RQ_1)*

Implementing the weight-optimized Gower's metric - as described by [26] - augments the CPCC by around 15% (from 0.84 to 0.97) if compared with the case where all weights are set to unity. We calculate the Pearson-γ correlation for our dataset to further investigate the benefits of our optimized clustering methodology. The results are shown in Figure 3.

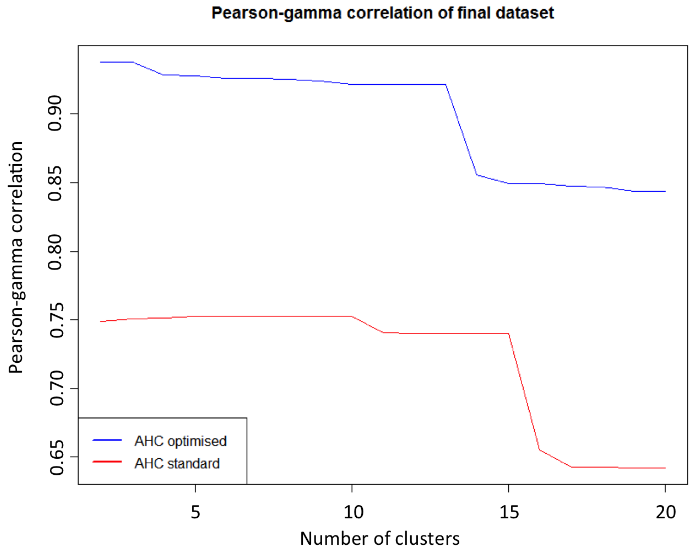

Figure 3. Pearson-γ correlation for the AHC - optimized (blue) and standard (red) - against the number of clusters.

Implementing the optimized weights for Gower's metric increases the Pearson-γ correlation by around 11%. That, together with the 15% improvement in the CPCC, indicates that our methodology to optimize weight works significantly better than the standard procedure. We note a slight decrease in the Pearson-γ correlation for the AHC optimized results at 4 clusters followed by a sharp decrease at 13 clusters. From the 13 clusters with a high Pearson-γ correlation, 4 main clusters contain around 98% of the total amount of unique users. Nevertheless, we include all 13 clusters in our analysis to ensure that these outliers do not influence these main 4 clusters.

4.2. *Prediction accuracy of the Cox proportional hazards model ($RQ_{2.1}$)*

In Figure 4 we show our results for the CPH model. We do so, by means of a ROC plot for four different time spans within the testing set.

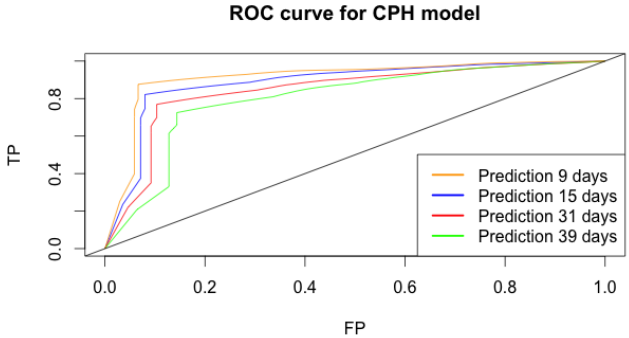

Figure 4. ROC curves for the CPH model based on the testing set. TP and FP indicate true positive and false positive, respectively. The legend indicates the different time spans.

We observe in Figure 4 the predictions for increasing time spans. As expected, the ROC curves approach the diagonal line (random prediction) as we move forward in time. Note that these predictions are based on the testing set, and not yet on the validation sets. That is because, at this stage, we are interested in the generalization capabilities of this model. We explain: these ROC curves are derived from the survival chance as a function of time. This means 100% survival chance for day 0, decaying eventually to 0% as time progresses (Kaplan-Meier curve). Based on the these probabilities, the ROC curves are generated within the testing set as an universal discrete prediction for the CPH model from 9 to 39 days. We see that both short- and long term predictions are accurate. The AUC ranges from 0.8 to 0.91 for 39 and 9 days, respectively.

4.3. *Prediction accuracy of the negative binomial model ($RQ_{2.2}$)*

Figure 5 presents the ROC curves for the NB model. Contrarily to the CPH model, the NB model predicts actions until disengagement. That means it would be fairly impossible to create a binary classifier able to estimate the exact number of actions before disengagement. Instead, we use 5 counts per 14 days as a threshold to determine if a user is engaged or not. In this case, a user is considered engaged if exceeding the threshold. Nevertheless, we note that the outcome is inferior compared to the results obtained by the CPH model. Due to the unexpected results for the testing set, we also analyze the performance of the NB model for the validation sets. The results, shown in Figure 5, remain reasonably similar to the ones obtained for the testing set. The AUC is fairly constant and equal to 0.67 for all the sets.

Figure 6 presents the number of events observed and predicted by the model to further understand the performance of the NB model.

Besides the fact that some of the predictions coincide with the observations, a very significant part of the observed values is crudely underestimated by the model. That means the model is able to reasonably predict the so-called “true positive” values but fails to predict the “true negative” ones. These results suggest that this model is, to a certain extend, accurately predicting the right counts to disengagement, albeit with many inaccurate predictions included as well.

4.4. *Prediction accuracy of the random forest model ($RQ_{2.3}$)*

In Figure 7 we visualize the ROC curves for the RF model applied to the different sets. We find that the AUC ranges from 0.93 to 0.83 for the *testing* and *validation 3* sets, respectively. The high AUC

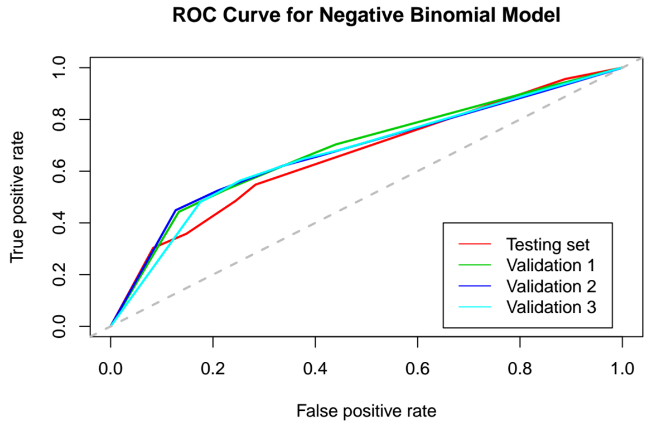

Figure 5. ROC curves for the NB model. The legend indicates the datasets. The timespan is fixed to 14 days.

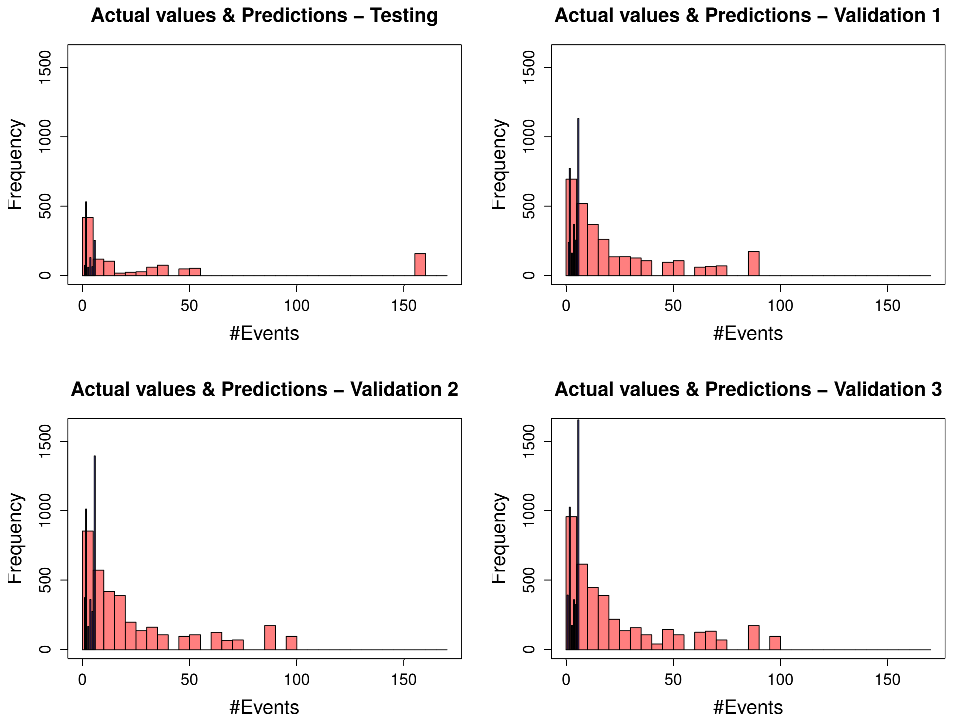

Figure 6. Comparison between the number of events predicted (black) and observed (red) for the different sets, as indicated in the headers. The timespan is fixed to 14 days.

values mean that the RF model is generic enough to classify our user as engaged or disengaged for all our dataset.

To further understand which processes/features determine the behaviour of this model, in Table 1 we show the mean decrease in accuracy (MDA) for some of the predictors. The MDA is calculated

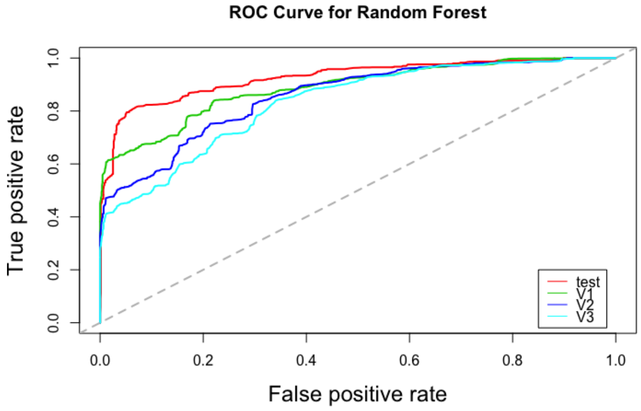

Figure 7. ROC curves for the RF model. The test set curve is shown in red, followed by the Validation 1,2, and 3 sets in green, blue, and cyan, respectively.

by permuting the values of each predictor and then measuring by how much the predictive accuracy decreases.

Table 1

Selection of predictors and their respective mean decrease in accuracy (MDA)

Predictor	**MDA** (%)
Groups	36.5
Number of actions	35.5
Longitude	34.0
Weekday	33.5
Latitude	27.0
Observation time	21.0

In our case, removing *groups*, *number of actions*, *longitude*, or *weekday*, from the predictors list would decrease the accuracy of this model by over 30%. We point out to the reader that the MDA is computed after the RF is trained. Therefore, training the model without these predictors will not drop the performance by the amounts shown in Table 1. Instead, the new model may find new correlated features unknown to the current model. We also notice in Table 1 the importance of adequately clustering users since *groups*, calculated with the optimized AHC algorithm, is responsible for the highest MDA value.

4.5. *Prediction accuracy of the XGBoost model ($RQ_{2.4}$)*

Figure 8 presents the XGBoost curves for the different sets. The AUC range is virtually the same as the one for the RF, with the values from 0.93 to 0.82 for the *testing* and *validation 3* sets, respectively.

To keep our comparison similar to that of the RF we have selected the same predictors and seen if there was any difference in their relative importance distribution. To calculate their importance we examined the “Gain” value. Interestingly, we see that the order of the importance remains the same as per the RF with “groups” being the predictor with the highest Gain value (0.06) and “obs time” with the lowest

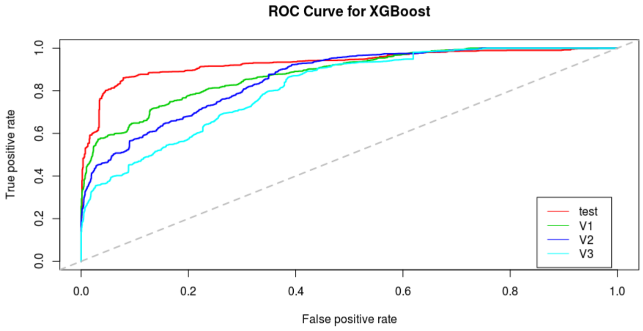

Figure 8. ROC curves for the XGBoost model. The test set curve is shown in red, followed by the Validation 1,2, and 3 sets in green, blue, and cyan, respectively.

(0.0001). That reinforces the importance of having well-defined and accurate groups as output from the clustering algorithm.

5. Discussion

Concerning RQ_1, the modified clustering algorithm containing optimized weights for Gower's metric performed adequately. The results showed an improvement of $\approx 11\%$ on the Pearson-γ correlation, and $\approx 15\%$ on the cophenetic correlation, if compared to a standard clustering methodology. The clustering outcome proved to be the most important predictor for both RF and XGBoost algorithms. That provides further motivation to optimize the clustering process aiming at sharpening the groups definition and as a consequence improve the machine learning results.

Concerning RQ_2, we applied the four models on the dataset and analysed the results obtained, mainly via the use of ROC curves. All models performed well, in their own right, with Cox proportional hazards, random forest and the boosted-tree models resulting in similar performance when predicting UE. The performance of the negative binomial model was not comparable to the other three algorithms.Most importantly, we concluded that under this framework we were able to better understand our observations.

As shown in Section 4, CPH, RF and XGBoost models result in similar values of accuracy. Their AUC values are similar, ranging roughly from 0.8 to 0.9. Our fourth model, the NB model, resulted in an AUC of 0.67. It is important to re-iterate that this AUC values should be taken as individual measures of performance and not used to compare models, as the manner of predicting and even the element of prediction is different according to the algorithm used.

Even with a high AUC score, there are still, however, a number of caveats concerning the generalization of the CPH model. More specifically, the results obtained with this model vary significantly for different sets of predictors. Interestingly, the good results found by the RF and XGBoost models can be partially explained by their generality. We will take advantage of this feature and use these models to "classify" UE in the future as well.

We are also interested to model re-engagement. Given the fact that the CPH model is unable to do so (since it predicts survival times), a Markov-like stochastic model becomes then a plausible replacement. The reason is that these models are able to provide the transition paths between engaged-disengaged and to obtain the rate parameter of these transitions. We emphasize that the RF and XGBoost models are also able to model re-engagement. In the near future, we aim to compare in detail the results obtained by the RF and XGBoost, with the transition model.

Finally, it is important to note that the accuracy we obtained in our evaluation is specific to the dataset related to the waste recycling app and cannot be directly transferred to other mobile apps or application domains. Indeed, the aim of this study is not to provide a general solution for all mobile apps in all domains, but rather, we focus on (i) providing evidence that it is possible to predict when app users are getting disengaged with good levels of accuracy and (ii) providing a reusable modeling framework for UE in mobile apps. Researchers and practitioners in application domains other than waste recycling can re-use our proposed framework and its underlying techniques, provided that they will be customized according to (i) the characteristics of their specific app domain (*e.g.,* a user of a social media app may be considered as disengaged after 1 day of inactivity, instead of 9 days) and (ii) the performance of the trained models (*e.g.,* in a different domain the negative binomial model may perform better than RF or XGBoost).

6. Summary and Future Work

In this study, we provided evidence that predicting when users of mobile apps get disengaged is possible with a good level of accuracy. We achieve this result by proposing and evaluating a framework to model and predict user engagement in mobile applications. The framework consists of a modified clustering model that serves as baseline for other four numerical models: (i) a Cox proportional hazards, (ii) a negative binomial, (iii) a random forest, and (iv) a boosted-tree algorithm. These models were trained and validated against an observational dataset obtained from a real waste recycling mobile application. Our results show that *in our case* both machine learning approaches (RF and XGBoost) are adequate to model user engagement for the considered app.

Analyzing user behaviour to predict and prevent disengagement certainly poses a significant challenge, both from the methodological and analytical points of view. Due to the complexity of this task, we limited this study to characterizing and evaluating our methodology to *predict* UE. In a follow-up study, we will investigate how to ultimately influence user behaviour by increasing re-engagement rates and decreasing disengagement. Moreover, further research will touch upon studying the re-engagement *process*. We then intend to use push notification information - extending on the work of [13] - to ultimately determine the most appropriate interaction for each user at any given time, aiming to augment usage (maintain engagement) and prevent disengagement. Understanding the role gamification plays in mobile apps is also crucial. It can be done by further investigating how people redeem their points earned (e.g. immediately after achieving a minimum threshold or after some accumulation). That information helps in determining the type of notification that can be sent to each user.

Acknowledgements

All the authors thank the Amsterdam Network Institute for partially funding this research and Coen Jonker for his remarks on the initial version of the manuscript. Eduardo Barbaro thanks Mobiquity Inc. for the support and the hours available throughout the research.

References

[1] A. Lella and A. Lipsman, The 2016 U.S. Mobile App Report, 2016, comsCore white paper.

[2] Statista, Number of available applications in the Google Play Store from December 2009 to June 2018, 2018. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[3] Android, Android developer portal, 2017. http://developer.android.com/about/index.html.

[4] Y.H. Kim, D.J. Kim and K. Wachter, A study of mobile user engagement (MoEN): Engagement motivations, perceived value, satisfaction, and continued engagement intention, *Decision Support Systems* **56** (2013), 361–370. doi:10.1016/j.dss.2013.07.002.

[5] P. Racherla, C. Furner and J. Babb, Conceptualizing the implications of mobile app usage and stickiness: A research agenda, *Available at SSRN 2187056* (2012).

[6] K. Wachter, Y.H. Kim and M. Kim, MOBILE USERS: CHOOSING TO ENGAGE, *International Journal of Sales, Retailing and Marketing* **1**(1) (2012). doi:10.5848/APBJ.2012.00002.

[7] J. Tang, C. Abraham, E. Stamp and C. Greaves, How can weight-loss app designers best engage and support users? A qualitative investigation, *British Journal of Health Psychology* **20**(1) (2015), 151–171. doi:10.1111/bjhp.12114.

[8] S. Snyder, Hyper-personalizing the User Experience through Data, 2016, Presentation at the Mobile World Congress - Contextual Commerce, Barcelona.

[9] J.A. Cafazzo, M. Casselman, N. Hamming, D.K. Katzman and M.R. Palmert, Design of an mHealth App for the Self-management of Adolescent Type 1 Diabetes: A Pilot Study, *J Med Internet Res* **14**(3) (2012), e70. doi:10.2196/jmir.2058.

[10] S. Snyder, *The New World of Wireless: How to Compete in the 4G Revolution*, FT Press, 2009, p. 208.

[11] S. Attfield, G. Kazai, M. Lalmas and B. Piwowarski, owards a science of user engagement (Position Paper), in: *WSDM Workshop on User Modelling for Web Applications*, 2011.

[12] Y. Liu and Y. Zhuang, Research Model of Churn Prediction Based on Customer Segmentation and Misclassification Cost in the Context of Big Data, *JCC* **03**(06) (2015), 87–93. doi:10.4236/jcc.2015.36009.

[13] A.S. Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber and A. Schmidt, Large-scale assessment of mobile notifications, in: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM Press, 2014. doi:10.1145/2556288.2557189.

[14] H.L. O'Brien and E.G. Toms, What is user engagement? A conceptual framework for defining user engagement with technology, *J. Am. Soc. Inf. Sci.* **59**(6) (2008), 938–955. doi:10.1002/asi.20801.

[15] H.L. O'Brien and R. Bassett, Exploring engagement in the qualitative research process, 2009, American Society for Information Science and Technology Annual Meeting, Vancouver, BC, October, 2009.

[16] K. Nelissen, M. Snoeck, S.V. Broucke and B. Baesens, Swipe and Tell: Using Implicit Feedback to Predict User Engagement on Tablets, *ACM Trans. Inf. Syst.* **36**(4) (2018), 35:1–35:36. doi:10.1145/3185153.

[17] J. Lehmann, M. Lalmas, E. Yom-Tov and G. Dupret, Models of user engagement, in: *Proceedings of the Conference on User Modeling, Adaptation, and Personalization*, UMAP, Springer, 2012, pp. 164–175.

[18] A. Ramesh, D. Goldwasser, B. Huang, H.D. III and L. Getoor, Learning Latent Engagement Patterns of Students in Online Courses, in: *AAAI Conference on Artificial Intelligence*, 2014.

[19] X. Wang, K. Zhao and N. Street, PREDICTING USER ENGAGEMENT IN ONLINE HEALTH COMMUNITIES BASED ON SOCIAL SUPPORT ACTIVITIES, in: *2014 INFORMS Workshop on Data Mining and Analytics*, 2014.

[20] H. Müller, J. Gove and J. Webb, Understanding tablet use, in: *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI*, ACM Press, 2012. doi:10.1145/2371574.2371576.

[21] P. Domingos, A few useful things to know about machine learning, *Communications of the ACM* **55**(10) (2012), 78. doi:10.1145/2347736.2347755.

[22] W.H.E. Day and H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, *Journal of Classification* **1**(1) (1984), 7–24. doi:10.1007/BF01890115.

[23] L. Anderlucci, Comparing different approaches for clustering categorical data, PhD thesis, University of Bologna, 2012.

[24] C. Henning and T.F. Liao, How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification, *Journal of the Royal Statistical Society* **62**(3) (2013), 309–369.

[25] J.C. Gower, A General Coefficient of Similarity and Some of Its Properties, *Biometrics* **27**(4) (1971), 857. doi:10.2307/2528823.

[26] J. van den Hoven, Clustering with optimised weights for Gower's metric, Master's thesis, Vrij University, Amsterdam, the Netherlands, 2016.

[27] S. Saraçli, N. Doğan and İ. Doğan, Comparison of hierarchical cluster analysis methods by cophenetic correlation, *J Inequal Appl* **2013**(1) (2013), 203. doi:10.1186/1029-242x-2013-203. http://dx.doi.org/10.1186/1029-242X-2013-203.

[28] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd edn, Springer, 1999.

[29] D.R. Cox, Regression Models and Life-Tables, *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(2) (1972), 187–220.

[30] D.R. Cox and D. Oakes, *Analysis of survival data*, 1st edn, Chapman and Hall, 1984.

[31] C.A. Bellera, G. MacGrogan, M. Debled, C.T. de Lara, V. Brouste and S. Mathoulin-Pélissier, Variables with time-varying effects and the Cox model: Some statistical concepts illustrated with a prognostic factor study in breast cancer, *BMC Medical Research Methodology* **10**(1) (2010), 1–12. doi:10.1186/1471-2288-10-20.

[32] P.J. Heagerty, T. Lumley and M.S. Pepe, Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker, *Biometrics* **56**(2) (2000), 337–344. doi:10.1111/j.0006-341X.2000.00337.x.

[33] A. Linden and S. Mantyniemi, Using the negative binomial distribution to model overdispersion in ecological count data, *Ecology* **92**(7) (2011), 1414–1421. doi:10.1890/10-1831.1.

[34] P. Buehlmann and B. Yu, Boosting with the L2 loss: regression and classification, *Journal of the American Statistical Association* **1**(98) (2003), 324–339.

[35] P. Buehlmann, Boosting for high-dimensional linear models, *The Annals of Statistics* **2**(34) (2006), 559–583.

[36] P. Buehlmann and T. Hothorn, Boosting algorithms: regularization, prediction and model fitting, *Statistical Science* **4**(22) (2007), 477–505.

[37] L. Breiman, Random Forests, *Machine Learning* **45**(1) (2001), 5–32. doi:10.1023/A:1010933404324.

[38] A. Liaw and M. Wiener, Classification and Regression by randomForest, *R News* **2**(3) (2002), 18–22. http://CRAN.R-project.org/doc/Rnews/.

[39] C. Goutte and E. Gaussier, A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation, in: *Advances in Information Retrieval*, D.E. Losada and J.M. Fernández-Luna, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 345–359.

[40] T. Chen and C. Guestrin, Xgboost: A scalable tree boosting system, in: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, 2016, pp. 785–794.

[41] T. Chen and T. He, Higgs boson discovery with boosted trees, in: *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, 2015, pp. 69–80.

[42] S. Dey, Y. Kumar, S. Saha and S. Basak, Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting (2016). doi:10.13140/RG.2.2.15294.48968..

[43] L. Torlay, M. Perrone-Bertolotti, E. Thomas and M. Baciu, Machine learning–XGBoost analysis of language networks to classify patients with epilepsy, *Brain informatics* **4**(3) (2017), 159.

[44] S.P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis and N. Vlachogiannakis, Forecasting stock market crisis events using deep and statistical machine learning techniques, *Expert Systems with Applications* **112** (2018), 353–371. doi:https://doi.org/10.1016/j.eswa.2018.06.032. http://www.sciencedirect.com/science/article/pii/S0957417418303798.

[45] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* **27**(8) (2006), 861–874. doi:10.1016/j.patrec.2005.10.010.