

# Understanding & Predicting User Lifetime with Machine Learning in an Anonymous Location-Based Social Network

Jens Helge Reelfs

Chair for Computer Networks  
Brandenburg University of Technology, Germany  
reelfs@b-tu.de

Oliver Hohlfeld

Chair for Computer Networks  
Brandenburg University of Technology, Germany  
hohlfeld@b-tu.de

Max Bergmann

Chair for Computer Networks  
Brandenburg University of Technology, Germany  
max.bergmann@b-tu.de

Niklas Henckell

The Jodel Venture GmbH  
Berlin, Germany  
niklas@jodel.com

## ABSTRACT

In this work, we predict the user lifetime within the anonymous and location-based social network Jodel in the Kingdom of Saudi Arabia. Jodel's location-based nature yields to the establishment of disjoint communities country-wide and enables for the first time the study of user lifetime in the case of a large set of disjoint communities. A user's lifetime is an important measurement for evaluating and steering customer bases as it can be leveraged to predict churn and possibly apply suitable methods to circumvent potential user losses. We train and test off the shelf machine learning techniques with 5-fold crossvalidation to predict user lifetime as a regression and classification problem; identifying the Random Forest to provide very strong results. Discussing model complexity and quality trade-offs, we also dive deep into a time-dependent feature subset analysis, which does not work very well; Easing up the classification problem into a binary decision (lifetime longer than timespan  $x$ ) enables a practical lifetime predictor with very good performance. We identify implicit similarities across community models according to strong correlations in feature importance. A single countrywide model generalizes the problem and works equally well for any tested community; the overall model internally works similar to others also indicated by its feature importances.

## CCS CONCEPTS

• **Networks** → **Online social networks; Location based services**; • **Computing methodologies** → **Classification and regression trees**.

## 1 INTRODUCTION

Every social networking platform depends on an active user-base. This user-base is threatened by *user churn*, which represents users leaving the platform. Retaining existing users is a core marketing strategy [9] to mitigate potential losses focusing on positive user relationships via data and behavioral analysis. Loyal users (possibly

inadvertently) advertise a product freely. More importantly, they tend being more profitable to a company. Beyond our field, customer lifetime value (CLV) denotes expected revenue over time in marketing and may be used to identify high-value and users at risk.

The actual churn prediction's goal is not only limited to predicting a churn event, but also the likelihood or time until a user might churn. Such individual churn probabilities allow for direct timed steering of single users (help, notifications, email) to improve individual retention. In the broader picture, this also allows for steering communities or customer-bases with the envisioned optimal features in mind: Maintaining a healthy user-base, which always is hoped to grow and converge into a well-mixed population. The prediction of user churn is a well studied data mining task (cf. Related Work § 7). However, these works predominantly focus on predicting churn in a single community typically represented by one platform. The degree to which they generalize beyond a single user-base is thus an open question. New types of location-based networks enable the establishment of many different disjoint communities within the same platform, e.g., enabling the study of information diffusion [12]. This location-based property forming multiple independent user bases within the same platform constrains thus enables the study of this currently open question on how churn models generalize beyond a single community.

In this paper, we analyze and predict user lifetime in Jodel, a mobile only *location-based* social media messaging app. Jodel establishes local communities relative to the users' location, thereby creating a multitude of individual local communities throughout a country. As users cannot communicate with others outside their community, their user-bases tend to be disjoint. This makes Jodel an interesting platform to study user lifetime in a large set of disjoint user groups / communities that are subject to the same application constraints while being comparable. Further, current user churn studies focus on non-anonymous networks for which social ties often are a contributing factor (cf. § 7). In turn, due to its anonymity Jodel enables us to study factors of user lifetime in the absence of such social ties or social credit. Our contributions are as follows.

- We study user lifetime in a location-based, anonymous social messaging application. Our goal hereby is creating prediction models for the lifetime of a user within a specified observation period. We leverage resulting models to implicitly show in-/equalities of these communities w.r.t. churn.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-8313-4/21/04.

<https://doi.org/10.1145/3442442.3451887>

- Among tested off the shelf machine learning algorithms, Random Forest provides the best results for predicting a users' expected lifetime, both in the case of a regression problem *and* a classification problem. Our models use two types of features: user and community. We observe the models to perform well for all communities.
- Creating a single countrywide model generalizes the problem and works equally well for any tested community; this overall model internally works similar to others as indicated by its feature importances. We argue that model feature importances can provide feedback for empirical patterns pictured by the envisioned ideal community and may help to better understand reasons for users to stay or leave a platform.
- At last, we use Random Forest to answer a supposedly simpler—and easier to answer—binary classification problem of practical relevance to network operators: Given an observation time period, will the users' lifetime be longer? This approach achieves even better prediction quality than any other presented classifier.

## 2 THE JODEL APP

Jodel<sup>1</sup> is a mobile-only messaging application which we show in Fig. 1. It is location-based and establishes local communities relative to the users' location (0). Within these communities, users can *anonymously* post both images and textual content of up to 250 characters length (3) (i.e., microblogging) and reply to posts forming discussion threads (4). Posted content is referred to as “Jodels”, colored randomly (3). Posts are only displayed to other users within close (up to  $\approx 20\text{km}$ ) geographic proximity (2). Further, all communication is *anonymous* to other users since no user handles or other user-related information are displayed. Only *within* a single discussion thread, users are enumerated to enable referencing to other users. Up to 1.500 threads are displayed to the users in three different feeds (1): i) *recent* showing the most recent threads, ii) *most discussed* showing the most discussed threads, and iii) *loudest* showing threads with the highest voting score (described next).

Jodel employs a community-driven filtering and moderation scheme to avoid adverse content. For an anonymous messaging app, community moderation is a key success parameter to prevent harmful or abusive content. In Jodel, content filtering relies on a distributed voting scheme in which every user can increase or decrease a post's vote score by up- (+1) (7) or downvoting (-1) (9), e.g., similar to StackOverflow. Posts reaching a cumulative vote score (8) below a negative threshold (e.g., -5) are not displayed anymore. Depending on the number of vote-contributions, this scheme filters out adverse content while also potentially preferring mainstream content. Further, every post can be flagged as abusive, which subsequently is displayed to voluntary, system-selected, community moderators that majority-vote to remove or keep the particular post. To increase overall user engagement in terms of creating content and voting, the network applies lightweight gamification by awarding *Karma* points (5).

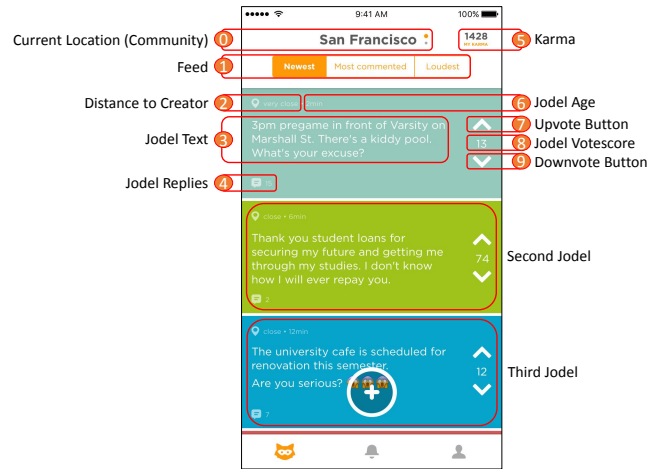


Figure 1: Jodel iOS mobile application.

## 3 USER LIFETIME

To study and model user lifetime, we first describe our dataset and then define how we assess a user's lifetime. The goal of our work is to automatically detect and classify the user lifetime in the anonymous and local communities within Jodel.

### 3.1 Dataset & Ethics

The Jodel network operators provided us with *anonymized* data of their network. The obtained data contains post and interaction *metadata* for 469M posts/replies created within the KSA and spans multiple years from from 2014 up to August 2017, while within KSA the application has experienced reasonable usage only since late 2016. For ethical reasons, it is limited to metadata only without textual content and user records stripped and anonymized. The data enables us to analyze individual users' interactions by their anonymous ID. Further, it contains no personal information and cannot be used to personally identify users. We inform and synchronize with the Jodel operator on analyses we perform on their data. The structure of our available dataset includes three object categories: interactions, content, and users.

- **Users** (about 1 M records) contains a user's accumulated karma value and whether the user is blocked.
- An **interaction** (about 1 B records) can be a *registered*, *post*, *reply*, *upvote*, *downvote*, or *flag*. Each interaction has a timestamp and a geohash. It is further linked to a user ID and a content ID.
- **Content** (about 469 M records) may either be a new post, i.e., starting a new thread, or a reply. This content records include a boolean flag whether it is text or an image (note: videos were added after our observation period).

**Dataset limitations.** Our dataset only includes the users' *active* interactions with the system, i.e., registering, creating posts, replying, or voting. Thus, we cannot infer when or how much a user only *passively* participates—lurkers—who only consume content. Further, the vote interactions are always mapped to the date and geoposition of the respective content creation. This prevents us from making

<sup>1</sup>Jodel, German for yodeling, a form of singing or calling. “Yudel” represents the adopted translation of Jodel to Arabic.

detailed analyses depending on the voting time or place. However, due to the vivid usage of the application within larger communities (multiple posts/replies per minute), we generally consider votes to be executed on the same day as their respective content. Especially since posts are only accessible via the three different feeds, where they will only stay for a very limited time, casting votes to content long after creation is usually not possible.

### 3.2 Goal: User Lifetime Prediction

**Definition.** We define the *lifetime* of a user as the time between the first (automatic account creation) and last system interaction (i.e., posting or voting) in minutes. Note that we can only define the lifetime by system interactions of a user since our dataset does not include passive activities (i.e., only reading). In prior work, churn, i.e., users leaving the system, is often defined as the end of a user’s lifetime. The lifetime enables us to partition users into timespan-dependent activity classes that we later predict, e.g., users that only used Jodel for a short amount of time, or longer.

class	lifetime	# users	fraction
1	0...1 days	135k	13.3%
2	1...7 days	123k	12.1%
3	7...14 days	75k	7.4%
4	0.5...1 month	124k	12.3%
5	1...3 months	268k	26.4%
6	> 3 months	288k	28.4%
$\Sigma$		1 012k	100.0%

**Table 1: The definition of six churn classes. We subdivide the user population by their active time.**

**User lifetime distribution.** We compute the lifetime for every user in our dataset and group users into six classes shown in Table 1. There is a broad range of different user lifetimes, ranging from users that only use Jodel for less than a single day and users that stick with the platform for more than 3 months. A user’s lifetime can end in two cases: First, the end of the observation period. Second, users that stop using the application (i.e., churn). As in a practical setting the observation period is always finite, prior work *approximates* the churn potential by using a threshold (e.g., no activity within the last  $n$  days, where  $n$  often is derived empirically); the finiteness naturally introduces a skew towards shorter time periods. If we apply such a threshold of one week  $n = 7$  (i.e., users are regarded as churners if there is no activity within the one week threshold margin towards the end of observation), 61% of the users will be defined as churners. We remark that churn prediction is an inherently hard problem since users could become active again after the threshold. Instead of churn, we predict user lifetime, i.e., the chance of a user to use the app for at least  $k$  days or a discretized timespan.

**Paper goal.** The goal of the paper is to predict the lifetime of a user within a specified observation period. Social network operators can use the resulting models as an online algorithm to predict the likelihood of a user to stick with the platform; furthermore, they allow for studying user behavior.

## 4 FEATURES

To predict the lifetime of a user by using a data-driven ML model, we derive features from *i)* the user itself and *ii)* her community.

**Engineering Subsets.** We introduce two different feature classes to represent the user and her environment: *i)* **User related features:** e.g., user registration event information, down-/upvoting and post/reply behavior of a specific user. *ii)* **Community related features:** e.g., posts/replies, up-/downvotes, average post response time of the users home community.

**Users home community.** Since Jodel establishes communities relative to the users’ locations, users can participate within different communities when sufficiently changing their geographic position. For a stable model, we derive the community features from the users’ home community, which we define as the city location with a user’s most interactions. For 87% of the users, this home community represents the city in which they initially registered. We use this attribute to determine a user’s city throughout this work.

**Capturing time.** While our features up to now do not catch any time-dependent information reflecting a user’s lifecycle, we add timing insights by duplicating the features with time-period bounds (1 day, 2 days, 3 days, 1 week, 2 weeks, 1 month, 3 months, >3 months). We end up with 66 (29) time-(in)dependent features. Noteworthy, none of typical scaling (e.g., std, min-max) or balancing (e.g., SMOTE, random over/undersampling) techniques improved any of our results significantly. Further, we used mean imputing as it provided best overall results.

## 5 RANDOM FOREST MODELLING

Within this section, we discuss our machine learning approach for a user lifetime predictor. By applying grid searches, after showing the Random Forest providing best results in comparison to other off the shelf ML methods, we dive deeper into the results of individual community models and their best parameters obtained. Then, we show how the prediction quality depends on model complexity and how different time-dependent feature subsets determine performance. Moreover, we look into model generalization and the impact of the amount of input data. Eventually, we derive implicit model (and as such community) similarities.

For any model, we apply grid-searches leveraging a random 5-fold crossvalidation approach for both, formulating a problem for *a)* regression of lifetime in minutes (REG), and *b)* classification into the previously defined classes (CLF). Our main evaluation metrics are: *REG:* The  $R^2$  score measures how equal real and predicted populations are, and *CLF:* the F1 score describes the harmonic mean of precision and recall; both providing an overall picture.

Since the property of establishing communities relative to a user’s position, our data set contains a large set of city-level communities throughout the country. To focus our discussion, in comparison to an all data *Country* model, we chose a subset of five cities varying in their size by the amount of users to create distinct prediction models: Riyadh, Jeddah, Mecca, Al Bahah, and Al Jafr (large to small). Our implementation uses Python scikit-learn off the shelf functionality.

## 5.1 ML Algorithm Selection

First, we ran a grid search for all data over a set of hyperparameters for standard ML methods to obtain ballpark numbers. These grid searches use a *mean* imputing strategy while not incorporating any scaling or balancing. Our used algorithms are Random Forest (RF), Decision Tree (DT), Multi Perceptron (MLP), AdaBoost (AdaB), K-Nearest Neighbors (KNN), and Stochastic Gradient Descent (SGD).

Table 2 provides an overview of a baseline in comparison to both problem formulations: regression and classification. The baseline is obtained by using the scikit-learn dummy regressor predicting the mean user lifetime in minutes, and the most frequent class for classification. While it yields unusable results for the regression task, an imbalanced dataset naturally provides better figures for classification. Due to multiple crossvalidation runs, we also provide standard deviation figures.

Algorithm	REG: $R^2 \pm \text{stddev}$	CLF: F1 $\pm \text{stddev}$
baseline	$-0.0000 \pm 0.0000$	$0.1672 \pm 0.0007$
RF	$0.9822 \pm 0.0004$	$0.9668 \pm 0.0005$
DT	$0.9580 \pm 0.0009$	$0.8049 \pm 0.0095$
MLP	$0.9668 \pm 0.0036$	$0.6768 \pm 0.0304$
AdaB	$0.7654 \pm 0.0012$	$0.6720 \pm 0.0093$
KNN	$0.6764 \pm 0.0013$	$0.5077 \pm 0.0006$
SGD	$0.3422 \pm 0.0551$	$0.1686 \pm 0.0191$

**Table 2: Off the shelf ML algorithm results using all data applying mean imputing; no scaling, no balancing. While the RF performs best, DT and MLP achieve similar regression performance falling short in classification.**

We observe that both, the regression and classification baseline are easily outperformed by any algorithm except for CLF with SGD and KNN. The best performing algorithm always is a rather complex Random Forest. However, for regression, DT and MLP also perform quite good. The results almost do not fluctuate across multiple crossvalidation instances at all.

**Findings.** *The best performing ML algorithm is the Random Forest with very strong regression  $R^2 \approx 0.97$  and classification F1  $\approx 0.99$  scores. Thus, we will from now on focus on the RF algorithm. Nonetheless, most others also outperform the baseline significantly.*

## 5.2 Independent Communities

As we have now determined the best-working algorithm for all data to be Random Forest and its parameters for our regression and classification predictor, we take a closer look into performance of specific independent community models. That is, does the prediction performance differ by community? This evaluation is enabled by the location-based nature of Jodel which allows us to compare independent user bases subject to the same platform constraints.

In Table 3, we show the best results of each Random Forest grid search instance for our selected cities and an all data country model. The communities are sorted by the amount of users within their community in descending order. We selected these particular examples due to their different amounts of users to cover a wide range from large to small. The  $R^2$  score describes the crossvalidation result

for the regression problem, whereas F1 describes the classification results; additionally, we provide the standard deviation across folds.

Community	REG: $R^2 \pm \text{stddev}$	CLF: F1 $\pm \text{stddev}$	#users
Country	$0.9822 \pm 0.0004$	$0.9668 \pm 0.0005$	1 012k
Riyadh	$0.9728 \pm 0.0013$	$0.9531 \pm 0.0006$	284k
Jeddah	$0.9667 \pm 0.0016$	$0.9372 \pm 0.0008$	101k
Mecca	$0.9551 \pm 0.0035$	$0.9185 \pm 0.0039$	45k
Al Bahah	$0.9457 \pm 0.0032$	$0.8752 \pm 0.0093$	11k
Al Jafr	$0.8219 \pm 0.1115$	$0.5807 \pm 0.0594$	174

**Table 3: RF classification and regression results for the Country and selected individual communities. Results are consistent and stronger for larger communities, except for Al Jafr due to small amount of data.**

Generally, all predictions perform well with  $R^2$  scores above 0.94 for all cities except Al Jafr; The F1 score within the classification instances is above 0.87 for most cases. There are negligible fluctuations across folds as seen by the low standard deviation for all cities again except Al Jafr due to the very few data points (only 174 total users); yet, the regression still works surprisingly well, whereas the classification falls short in achieved quality. We observe that all predictors work better on larger communities, hence more data.

Further, besides our best performing Random Forests mostly use larger amounts of estimators, we notice that they are complex being rather deep and leveraging all features ( $n$ ). We will discuss tree complexity versus performance in Subsection 5.3.

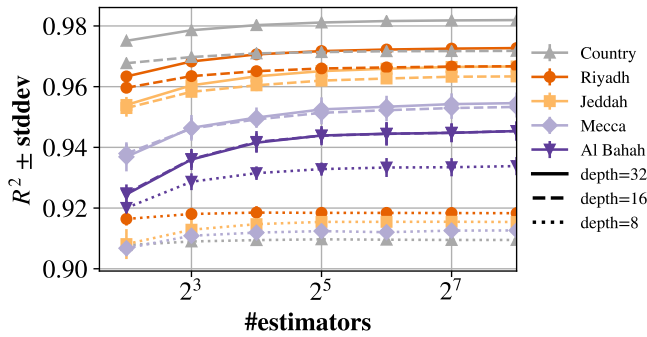
**Findings.** *The overall predictor performance for the regression and classification task is very good for all analyzed independent communities except Al Jafr. Resulting models are quite complex in terms of tree depths, used features and estimators. Best-performing classifications tend to require less complex model instances than the regression.*

## 5.3 Predictor Sweet Spot

As we have presented best performing results from the grid search for the Random Forest across our selected communities, we now want to shed light on the relationship between model complexity and prediction quality for two reasons: 1) An overly complex model might tend to overfit our data, 2) Less complex models are usually preferred due to less computation times for both, fitting and application.

We therefore investigated the relationship between used features, estimators, tree depth and resulting model quality for both, CLF and REG. For both problem formulations, model complexities are qualitatively very similar to their achieved performance—expectedly, more complex models perform better.

Exemplary, we show the relation between used estimators, tree depth and quality for the regression model using all features in Figure 2, which ultimately allows us to define a sweet spot. The logarithmic x-axis denotes the amount of used estimators (tree instances of the ensemble), whereas the y-axis shows the resulting  $R^2$  score and standard deviation as error bars. There are three series for each city for a tree depth of 8, 16, and 32, respectively; we removed Al Jafr to increase readability.



**Figure 2: RF regression performance vs. model complexity.** More complex models provide stronger performance with diminishing returns in depth and especially the amount of used estimators.

We observe that increasing the tree depth substantially increases model quality. However, the improvement from a depth of 8 in comparison to 16 is by far larger than the change from 16 to 32. For the smallest community shown, Al Bahah, increasing the tree depth above 16 does not improve performance. While 32 estimators already yield very good results, the quality increase is asymptotically bounded, i.e., there are diminishing returns.

**Findings.** *Although our grid search shows best results with rather complex Random Forest models (cf. § 5.2), our in-depth analysis of hyperparameters vs. quality reveals that only few estimators with mediocre tree depths already yield very good results with diminishing returns of increasing model complexity.*

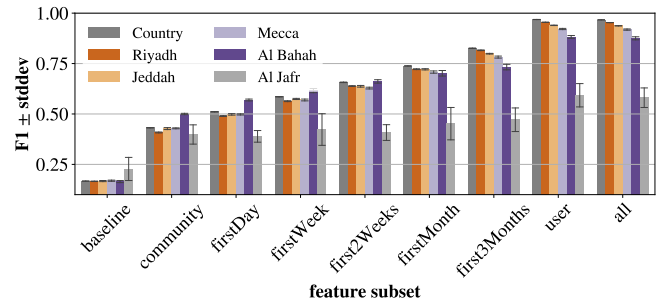
#### 5.4 Feature Subset Analysis

While our prediction for regression and classification works well, we next want to determine the impact of different feature subsets. That is, which feature subset provides best results, or even better results than using all features? We conduct grid searches for the Random Forest across community and user features for all data (Country) and each of our selected independent communities.

Furthermore, from a practical implementation standpoint, one might think about leveraging a sliding window approach over a single day, a week or longer as model input. Such an approach makes a model more time-invariant. However, limited knowledge may seriously impact the model performance—as it is expected to degrade.

Thus, we show the impact of our feature subsets within Figure 3 on the classification example. Note that the subset impact is quite similar for the regression problem (not shown). The x-axis describes used feature subsets: community and time-dependent features being cumulative: E.g., the *firstWeek* subset also includes features for the shorter time periods of *firstDay* and *first3Days*. The y-axis denotes the model quality via the F1 metric and the standard deviation across folds.

First of all, we observe that the community features alone provide worst results, but are quite similar to only looking into user data of her individual first days at scores ranging from  $\approx 0.4 \dots 0.6$ . By increasing the observation window up to 3 months, model quality increases drastically to F1 scores  $> 0.8$  for most cities. Only relying



**Figure 3: RF Classification feature subset performance results.** With increasing observation time per user, the results improve significantly. Overall, the country model works best, while most others achieve similar performance.

on user features provides similar results to using all features; the community features by itself only have a negligible impact on prediction quality. To be clear, this does not imply that the community only has negligible impact on user lifetime or user experience.

Noteworthy, predictions for larger communities tend to be better than for e.g., the Al Jafr community, always being off presumably due to its very few users (only 174). We cannot explain why the predictions for Al Bahah perform better for short time frames.

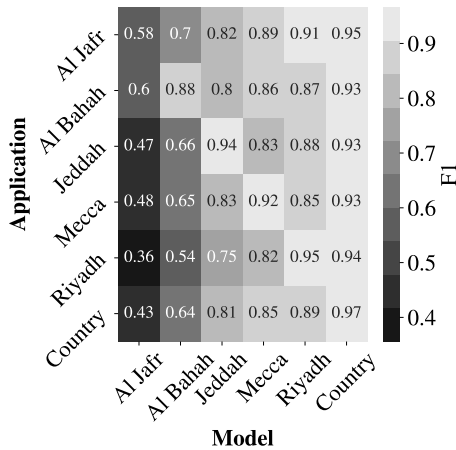
**Findings.** *Although our prediction models have proven very good performance, taking a practical stance by only using timely-windowed features depending on the users' active time reveals that classification and regression (not shown) quality deteriorates for real-world use-cases.*

#### 5.5 Generalization

We next study how well our models predict the lifetime from other communities to investigate whether there is a model—possibly a community, or the Country-model—providing well-suited allround prediction quality.

Within Figure 4, using the best-performing model from our previous grid-search (cf. § 5.2) each, we provide cross-application (community-model  $\times$  predictions-for-community) classification scorings of our different community and the Country model(s). The x-axis describes the used model instance, whereas the y-axis denotes the predictor input-dataset. We provide the macro F1 scores for each combination, colored on the z-axis. Note that we added the same-community model/application F1 scores from previous results as a baseline (diagonal upper left to bottom right).

Focusing on the *Country*-model first, *i*) we observe that it provides strong generalized performance with F1 scores  $\geq 0.93$  throughout any community (rightmost column). Taking a closer look into the community cross applications, we find: *ii*) Most individual community models perform very well on their own input dataset; other models significantly improve the Al Jafr community prediction scores. *iii*) The overall best-working community model is Riyadh, falling short in prediction quality for Mecca. The community models from Mecca (Jeddah) still delivers acceptable prediction quality across the board with F1 scores  $\geq 0.82$  ( $\geq 0.75$ ). However, the models for the smaller communities do not perform well in a generalized setting. *iii*) By leveraging the model scores as a proxy for



**Figure 4: RF classification model cross application results. We used each created model and performed a prediction for every other dataset. The diagonal same-same instances depict earlier prediction results as a comparison. The country model works best across the board.**

community similarly w.r.t. user lifetime, we identify that due to shown options for generalization, user lifetime is similar to some extent across the analyzed communities, yet through the models only being implicitly defined. iv) For the regression use-case, we find the Country model strictly outperforming all others with an  $R^2$  score of 0.98 in all communities; besides, only the Mecca community model provides consistent strong results applied to other communities at  $R^2$  scores  $\geq 0.83$  (not shown).

This generalization reveals that the independent communities can be captured well in a single model and behave similar to some extent w.r.t. user lifetime. Note however, the *Country*-model may be skewed in favor of larger communities.

**Findings.** *The overall Country model performs very well throughout any tested community for both, regression and classification, with improvements for smaller communities and (for classification) slight deteriorations for the clique of larger cities (Riyadh, Jeddah, Mecca). Yet this model works well and might be used for the whole dataset as a unique predictor, retraining is computationally heavier than selected individual models due to its size.*

## 5.6 Country Model in Detail

Our evaluation and cross application showed that the *Country*-model provides all-round performance for regression and classification, while also improving predictions for communities with comparably fewer users. But why is that? Does simply the amount of available data improve the model, or does the country model represent a better cut through the population?

To answer this question, we randomly downsampled all data to the reference values of our other selected communities and ran grid-searches for these new sampled Country models. We present our results in Table 4 for the classification problem. The alike community column depicts the reference sample size, whereas the F1 country column denotes the model’s classification quality. Further,

#	alike	F1 country $\pm$ std	F1 comm. $\pm$ std	$\Delta$
0	Country	0.9668 $\pm$ 0.0005	-	-
1	Riyadh	0.9496 $\pm$ 0.0016	0.9531 $\pm$ 0.0006	-0.0035
2	Jeddah	0.9312 $\pm$ 0.0026	0.9372 $\pm$ 0.0008	-0.0060
3	Mecca	0.9136 $\pm$ 0.0024	0.9185 $\pm$ 0.0039	-0.0049
4	Al Bahah	0.8773 $\pm$ 0.0063	0.8752 $\pm$ 0.0093	0.0021
5	Al Jafr	0.6794 $\pm$ 0.0691	0.5807 $\pm$ 0.0594	0.0988

**Table 4: RF classification Country model with limited input data corresponding to other city sizes. The country model achieves similar performance to individual models when restricting the input data size. I.e., by reducing the amount of data, performance deteriorates.**

we add the individual city models as an expected upper-bound baseline comparison (column F1 city).

Our evaluation reveals that the model performance remains very strong, but worsens with less data. However, given a statistical significance due to the sheer amount of input data, the observed delta to the individual model arguable remains within margin of error; except for Al Jafr most probably due to its very sparse data.

**Findings.** *We observe that the country model performance is similar to the individual models provided it uses the same amount of input data. Still, the country model deals better throughout all communities than any other individual model.*

## 5.7 Feature Relevance per Community

We observe the Random Forest to provide good prediction results. Further, we have seen that the communities behave similar w.r.t. user churn to some extent. However, we are missing information which metrics were predominantly used by the models. Do they rely on the very same features or have they learnt differently?

There are different well-known feature importance predictors, such as Relieff or RFECV. Here, we focus on RF Feature Importance (RFFI), Gini importance/Mean Decrease in Accuracy, depicting the percentage ranking within its decision making. By cross correlating the RFFIs with the ranked Spearman’s, we next want to figure out if importance score line up across the different models. We show the results for the classification in Figure 5, showing a community on both axes, of which the correlation between the cities’ importance vector is represented textually on the also colored z-axis. Note that the overall picture remains the same for Regression (not shown).

We observe that the clique of Mecca-Jeddah-Riyadh line up quite well, whereas the smaller cities fall off. Interestingly, the Country model feature importances also correlate to beforementioned clique indicating that these models work similarly. Digging deeper into the Country model by correlating importances of the sampled versions (not shown), we find that all of these models, despite using less data, are strongly correlating in feature importance and thus, share similar internals; except alike-Al Jafr falling short.

**Findings.** *The rather strong correlation among the models for larger cities strengthens our hypothesis of similarly behaving communities due to their model similarities. This also holds true for the overall Country model and its sampled versions.*

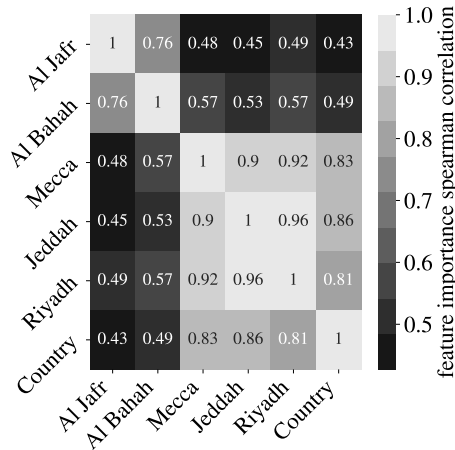


Figure 5: RF classification features importance spearman correlation. We find that most models strongly correlate in feature importances indicating similar model internals.

### 5.8 Future Work: Empirical Lifetime Study

Having built well-performing predictors for user lifetime and having seen that model feature importances often correlate, these importances figure an important signal of usage within the respective model, which deserve to be fed back into a thorough empirical analysis to better understand model internals.

We argue that used feature values of users subdivided into their lifetime classes represent a projection of its population w.r.t. user lifetime. I.e., by partitioning a feature’s population by the optimization target (active time), we discretize the community state available to each user. Implicitly driven through lifetime predicting RFs, a community state for important single features w.r.t. lifelime might then be given by the distribution in a certain time-slice—which in turn depict qualitative differences between these partitions.

However, while punctual information is not sufficient to identify changes, we are more interested in qualitative changes over time within most important features. To provide empirical insights, we first removed upper and lower 1% outliers and then applied Min-Max scaling (onto 0 to 1). Then we calculated the set of quantiles 10%, 20%, . . . , 90% and plotted the resulting kernel for each feature over the time subsets (1 day, 3 days, 1 week, 2 week, 1 month, and above) in Table 5, while the solid line denotes the median. That is, the (second) outmost gray area denotes the population between the quantiles of 10% (20%) and 90% (80%). To better capture qualitative changes, we apply a logarithmic y scale.

Given new challenges like for some metrics, we can observe clear trends, while others draw an unclear picture, or any counting feature depending on time, we leave next steps for future work. Nonetheless, feature candidates with clear trends might be a valuable object for future research, i.e., learning from model behavior; and ultimately e.g., testing hypotheses with e.g., synthetic tests [1].

## 6 BINARY LIFETIME PREDICTION

Although our prediction works quite well overall with classification F1 scores up to 0.95 for Riyadh, having only timely limited user

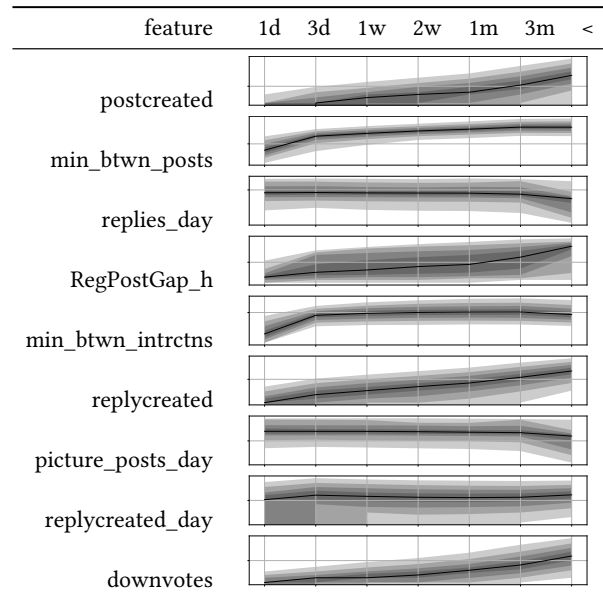


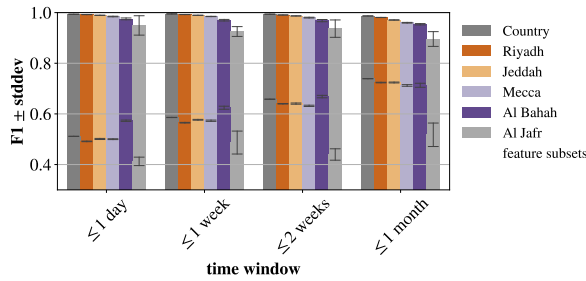
Table 5: Qualitative population of most important features (country model) subdivided into user lifetime sets. The areas denote quantiles from 10-90%, 20-80% etc. whereas the solid line depicts the median. On a logarithmic y-axis, we observe drastic variations within the metrics over a user’s lifetime.

information deteriorates prediction quality significantly (cf. § 5.4). In practice, e.g., a network operator usually only asks whether a user is likely to churn in near future. This allows us to reformulate our problem into a supposedly simpler—and easier to answer—binary classification problem: Given an observation time period, will a user’s lifetime be longer?

Thus, we ran grid searches for binary predictors for every lifetime class in Table 1 and all selected communities likewise to the feature subset analysis, which allows us to generate predictions according to our chosen time-periods of the feature subsets.

We show the results of the best binary classification models in Figure 6. The x-axis describes the used time window, whereas the y-axis denotes the F1 score. The model results for each independent community are plotted as bars with whiskers indicating the standard deviation across folds. For comparison, we added feature subset analysis results (hatched).

We first observe that the binary classification model works quite well with F1 scores above 0.95 for almost all communities across the time window. The performance delta to the cumulative time-dependent feature subset analysis models only becomes smaller for longer time frames as those models improve. With regards to model complexity, we observe similar results as seen for the other models (cf. 5.3): That is, e.g., a tree depth of 16 performs far better than 8, but there are diminishing returns beyond this depth and more than 16 estimators. In summary, the binary practical classifier outperforms any other presented classification model.



**Figure 6: RF Binary Classification performance results. Regardless of observation time, the binary lifetime prediction works exceedingly well. All time-dependent feature subset comparisons perform by far worse.**

**Findings.** *In practice, it is desired to remove complete time-dependency. Thus, the overall models may not reflect real-world performance in windowed feature use-cases. We however showed that classifications on time-dependent window subsets do not perform well. To ease up this problem, we simplified the task to a binary classifier predicting a users' lifetime. This approach achieves better prediction quality than any other presented classifier.*

## 7 RELATED WORK

User churn prediction has been a research topic for decades, yet with new emerging use-cases and technical advancement in data mining, machine learning and explainable AI, research on this topic has not halted by any means. We have seen various settings and applications within, e.g., telecommunication [17], social networks [4], online/video gaming [3, 13], or online marketing [2]. User lifetime has also been modelled as a survival analysis [16].

While user churn prediction often describes a binary classification, users' retention time might also be of interest. From a marketing perspective, user churn measures are typically weighted into an optimization target of a Customer Lifetime Value (CLV) according to, e.g., profit, yet relying on the same building block [2, 7].

Early work focussed less on today's off-the-shelf ML techniques. That is, statistical modelling and distribution fitting has shown significant success [4, 6, 8]. Besides applying Markov models [13], others have evaluated, e.g., evolutionary [14] or relational [8, 17] mining techniques. Nonetheless, various classical ML approaches have shown promising to very strong results with, e.g., boosting [10], DTs & tree ensembles [5, 6, 11]. Neural networks have also been applied to the problem in various architectures: E.g., deep [2, 15] or convolutional [3] NNs. Explicit feature engineering for the data-driven methods requires an individual process to the very field of application. However, research suggests that social ties and graphs are an important information carrier [6, 15]. Some research adds specific building blocks into their ML pipeline, such as user embeddings from browsing sessions [2]. Although, e.g., RF or DT importances are often given, there is lack of its discussion, i.e., backfired empirical implications are seldomly drawn. Explainable AI is currently more or less tackled by, e.g., applying a user pre-clustering [15], however.

## 8 CONCLUSION

In this paper, we analyze and predict the lifetime of a user in Jodel, a mobile only location-based messaging app. Our results show that Random Forests models provide good prediction results for both, regression and classification tasks across a selection of individual communities of varying sizes throughout the Kingdom of Saudi Arabia. When making models invariant to total observation time, i.e., only relying on timely limited feature sets, prediction results deteriorate substantially. This can be solved by using Random Forrest models to predict a simpler binary classification problem of practical relevance to network operators: Given an observation time period, will the users' lifetime be longer? This approach achieves even better prediction quality than any other presented classifier.

The location-based nature of Jodel yields the creation of disjoint communities throughout the country. When training a single model to the entire data set (i.e., a country-wide model), this model performs well compared to individual community models at similar amounts of input data. That is, while the individual communities are disjoint, users share behavioral pattern. This is further highlighted by the fact that the RF feature importances correlate between most individual and the country model(s). We therefore conclude the models' internal decision-making processes being similar and hence, also communities sharing alike behavior w.r.t. user lifetime.

Eventually, we argue that the feature importance provides strong hints about model internals and are a good starting point to be fed back into empirical analyses, which we leave for future work.

## REFERENCES

- [1] Muhammad Amjad, Vishal Misra, Devavrat Shah, and Dennis Shen. 2019. mRSC: Multi-Dimensional Robust Synthetic Control. *Measurement and Analysis of Computing Systems* (2019).
- [2] Benjamin Paul Chamberlain, Ângelo Cardoso, C. H. Bryan Liu, Roberto Pagliari, and Marc Peter Deisenroth. 2017. Customer Lifetime Value Prediction Using Embeddings. In *SIGKDD*.
- [3] P. P. Chen, A. Guitart, A. F. del Río, and Á. Periañez. 2018. Customer Lifetime Value in Video Games Using Deep Learning and Parametric Models. In *Big Data*.
- [4] Xi Chen, Ruibin Geng, and Shun Cai. 2015. Predicting Microblog Users' Lifetime Activities - A User-Based Analysis. *Electron. Commer. Rec. Appl.* (2015).
- [5] Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities. In *WWW*.
- [6] Gideon Dror, Dan Pelleg, Oleg Rokhlenko, and Idan Szpektor. 2012. Churn Prediction in New Users of Yahoo! Answers. In *WWW-CQA*.
- [7] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. 2005. RFM and CLV: Using iso-value curves for customer base analysis. *Journal Marketing Research* (2005).
- [8] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. 2005. "Counting your customers" the easy way: An alternative to the Pareto/NBD model. *Marketing science* (2005).
- [9] Philip Kotler. 2016. *A framework for marketing management*. Pearson Education.
- [10] Aurélie Lemmens and Christophe Croux. 2006. Bagging and Boosting Classification Trees to Predict Churn. *Journal Marketing Research* (2006).
- [11] Jagat Sastry Pudipeddi, Leman Akoglu, and Hanghang Tong. 2014. User churn in focused question answering sites: characterizations and prediction. In *WWW*.
- [12] H. Reelfs, T. Mohaupt, O. Hohlfeld, and N. Henckell. 2019. Hashtag Usage in a Geographically-Local Microblogging App. In *LocWeb (WWW)*.
- [13] J. Runge, P. Gao, F. Garcin, and B. Faltings. 2014. Churn prediction for high-value players in casual social games. In *IEEE Computational Intelligence and Games*.
- [14] Wai-Ho Au, K. C. C. Chan, and Xin Yao. 2003. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Evolutionary Computation*.
- [15] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. 2018. I Know You'll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application. In *SIGKDD*.
- [16] África Periañez, Alain Saas, Anna Guitart, and Colin Magne. 2016. Churn Prediction in Mobile Social Games: Towards a Complete Assessment Using Survival Ensembles. In *IEEE DSAA*.
- [17] María Óskarsdóttir, Cristián Bravo, Wouter Verbeke, Carlos Sarraute, and et al. 2020. Social Network Analytics for Churn Prediction in Telco: Model Building, Evaluation and Network Architecture. *Expert Systems with Applications*.