What Reveals Your Exact Age in Online Social Networks?

Nicolas Schäfer Samim Zahoor Taray Kwabena Amponsah-Kaakyire

Abstract

Online social networks have seamlessly made their way into the lives of the people. Users of OSNs choose to share certain information about themselves publicly. We show that with sharing this information the user can unintentionally reveal their age. From the raw information like profile fields and friendship relations we construct features and evaluate which features, if revealed, pose a risk to user privacy in the context of age. We perform experiments on a real world online social network and predict age of users within 2.62 years on average, 36% better than the previous work. For a more insightful analysis and comparison of regressors on imbalanced datasets we come up with more expressive metrics of evaluation.

1. Introduction

Online social networks provide platforms for their users to publicize their personal information. Based on whether a user considers some information about themselves sensitive or non-sensitive, they choose to publish it publicly online or not. Our study aims to show that it is possible to reveal information that a user might consider sensitive (and hence choose not to reveal it) from information that is made public by the user. We perform our study on the Pokec dataset [5] which consists of profile and friendship data from a Slovakian online social networking platform. We target the *age* of users and aim to find which other information is most relevant to accurately predict the age of a user. We extract useful features from raw information given in the form of profile data and friendship data in the dataset. We manually filter and encode the profile data to get useful numerical and categorical features from it. For the friendship data we use node2vec [4] to embed each user in the dataset into continuous vector space of fixed dimensions. We use the features to train machine learning models like Linear Least Squares, Gradient Boosting and Fully Connected Neural Networks for exact age prediction. We make an assessment of which features are especially useful for prediction, what reveals which features are more privacy critical in the context of age prediction and should be treated more carefully by the user. We find that common metrics used to evaluate the performance of age prediction methods in online social networks on imbalanced datasets have many shortcomings. For example the distribution of users in our dataset is imbalanced (users in the range 15-30 make up most of the samples) and metrics like Mean Absolute Error (MAE) either do not give the full picture or can even be misleading. We come up with expressive metrics that give more insight on model performance.

2. Related Work

[8] perform exact age prediction on the Pokec dataset. They use Deepwalk [7] on the network graph to embed each user into a continuous vector space and use the embedding to train a simple least squares regression model. They do not use profile data from Pokec and as we will show this provides significant improvements towards exact age prediction. Our analysis of features extracted from profile and friendship information reveals privacy risks to age information of the users and we also argue about the measures to counter these risks. [6] perform age prediction using messaging data. [9] analyze the content published by users on Twitter and try to predict age of the users. Both of these works treat age prediction as a classification task and try to classify users into age brackets, we design the task of age prediction as continuous regression problem.

3. Dataset

Pokec is an online social networking website [1] in Slovakia. We use the Pokec dataset [5] which consists of profile information and friendship information of the users of this OSN. The dataset was collected by crawling of the website by the authors of [5] and saving all information that was made public by the users. Specifically this information includes

- Profile information that the users made public like gender, region, hobbies, height, weight, marital status, age etc.
- A graph of the entire social network representing the friendship relation among the users.

In total around 1.6 million users (vertices) and 40 million relations (edges) are published in the dataset. Around 69% of the users make their age public. The age distribution is shown in Figure 1. The mean age is 24.94. The distribution is imbalanced and is dominated by people lying in the range of 15-30 years old. Using a constant regressor returning the mean can give seemingly good results (based on mean absolute error) because most of the users in the dataset fall around the mean value. Thus care has to be taken in interpreting prediction performance of a model using common metrics like mean absolute error. We address this with a specific metric for evaluation in section 6.

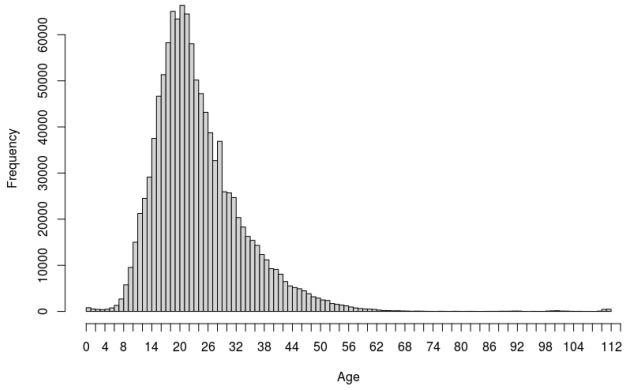

Figure 1: Histogram showing the age distribution of the dataset

4. Preprocessing and Feature Extraction

The user profile information consists of 59 raw fields. The type of values stored in a field varies. This can be a boolean, a number or a text description (most of the fields are text). We categorize the fields based on the type of values stored in that field as follows

- Boolean Fields: These are the *gender and public* fields with only two possible values.
- Integer values: These are the *completion percentage, age* fields. The *age* field gives the age of a user as an integer on the date of data collection.
- Text Fields: Most of the fields belong to this category and contain descriptive texts provided by the user. Examples are *marital status, height, weight, education, smoking*

Most of the fields in the profile data fall in the "Text Fields" category. Such data cannot be used in models for prediction or for correlation analysis directly and has to be encoded into a suitable form. We restrict ourselves to manual feature extraction and encoding techniques and do not consider text embeddings, since encoding text into an embedding can capture more data than the actual intention of the feature and so our results are not necessarily transferable to other datasets or OSNs where other descriptive text is given or only categorical data is given. Considering the feature "marital status" it is possible, given the embedded version, the feature can be identified as carrying useful information for the prediction of age. However, this could be mainly because of additional information that was part of the descriptive text of marital status and is captured in the embedding and not because of the actual marital status. In another dataset the feature could be categorical not providing any additional text. So our results would not be transferable. We try to give general statements about the relevance of features and so encode these manually to capture the actual meaning of the feature. We perform an initial filtering of the fields based on the difficulty and effort required for manual encoding and narrow down the set of text fields we encode to *completed level of education, smoking, marital status, height, weight*. We describe our methods for extracting useful features from the raw information in these fields next.

4.1. Feature Encoding

Unformatted string fields (e.g. marital status, completed level of education) are converted to categorical features using the following algorithm.

1. For each feature we find sensible keywords that correspond to a category.
2. For each sample of a feature we check if it contains a keyword and if it does, it is mapped to the corresponding category.

Finding Keywords:

1. We build a histogram of sub-strings consisting of n consecutive words for n between 1 and 3.
2. These sub-strings are then ordered by the histogram count and sub-strings with a count below a certain threshold are removed. This threshold is defined after manual inspection.
3. We inspect the remaining sub-strings manually to verify the result and remove sub-strings that only contain uninformative words (*e.g.* "is", "and"). We then map sub-strings to categories based on their meaning. Several sub-strings can have the same meaning and thus are mapped to the same category.

4. We order the final sub-strings in descending order by length to prevent that sub-strings that are contained in other sub-strings are not considered first when we remap (e.g. "not smoking" must be considered before "smoking" otherwise all "not smoking" samples are mapped to smoking)

5. The resulting list is the list of keywords where we consider a sub-string as a keyword mapped to some category.

Remapping: For each sample we check in order of the keyword list if any keyword exists as a sub-string in the sample. If it does, we map this sample to the category corresponding to the matching keyword. If we cannot match any keyword, the sample is removed.

Height and Weight: Inspecting the values in the height and weight field reveals that they are paired with the units 'cm' and 'kg' respectively. We employ pattern matching and extract the numerical value corresponding to 'kg' which becomes the value of the weight feature and 'cm' which becomes the value of the height feature.

4.2. Removing Outliers

We follow a conservative approach for outlier detection and removal. We tag samples as outliers only in the cases where we are absolutely sure that the values cannot occur in the real world. We observe some obvious outliers in the height and weight fields having impossibly large values and we remove these samples. For the age field, the dataset contains users 1 to 110 years old. It is unlikely that a 2 year old maintains a profile but we still consider it as a valid user because the profile can be maintained by their parents on their behalf. Same is the case for users with age more than 100.

4.3. Graph Embeddings

Pokec dataset contains a graph of the entire social network representing the friendship relation among users. We use node2vec [4] to extract n-dimensional graph embeddings from this graph. Node2vec is a framework for learning continuous n-dimensional representations for nodes in networks. The mapping of nodes to n-dimensional vectors is learned by maximizing the likelihood of preserving network neighbours of a node. The mappings for a user are 'similar' to the mappings of the user's 'neighbours' in the network. Such a representation is especially useful for the task of age prediction because of social homophily (age of people is similar to that of their friends). The hyperparameters for Node2vec that we use are: dimensions of the learned feature vectors $d = 128$, length of walks per source $l = 80$, number of walks per source $n_0 = 10$, return parameter $p = 1$ and Inout parameter $q = 1$.

4.4. Final Dataset

Finally we remove the samples with missing fields. We use one-hot encoding for categorical features. In our case numeric features differ significantly in scale so we perform min-max normalization for numeric features so that all features have values in the range $[0, 1]$. We end up with 221447 samples. Our pre-processing does not bias the dataset as the age distribution is almost the same after all the processing. The final feature that we work is shown in Table 1.

Gender
Marital Status
Height
Weight
Completion Percentage
Last Login Date
Registration Date
Completed Level of Education
Smoker
Public
128 Dim. Graph Embeddings

Table 1: Final List of Features

5. Correlation

To investigate linear relations between each predictor and age we consider the R^2 metric to get an approximated indication about the relevance of each predictor. We will use these results in section 6, where we build regression models for age prediction.

In the one dimensional setting, R is equal to the Pearson's correlation coefficient, but R and R^2 are even applicable in the multidimensional case. Because of that we can investigate quantitative and qualitative predictors (which we encode as one-hot vectors), both with R^2 as the metric.

R^2 is a measure for the explained variance in the output (here age) that is predictable from the input (other features) and its value lies between $[0, 1]$. 1 indicates a perfect linear fit and implies that there is a linear relationship between the output variable and the predictor.

From the results in Table 2, we do not see a predictor that strongly correlates with age. Weight, marital status and the graph embedding features show the best results. A higher R^2 value can be used as an approximated indicator for containing more useful information about the output. It is worth noting that predictors that have a strong but highly nonlinear relation are underrated in this metric and predictors that do not have a high R^2 value are not necessarily useless. In combination with other features they can reveal useful information. That is why we will treat these results carefully in section 6.

	R^2
public	0.0167
completion percentage	0.0691
gender	0.0072
last login date	0.0082
registration date	0.01591
height	0.0494
weight	**0.1573**
completed education	0.0623
smoking	0.0189
marital status	**0.2688**
128 dim. graph embedding	**0.5782**

Table 2: R^2 for each predictor predicting age in a linear least squares model

6. Prediction

We design the task of age prediction as continuous regression task, because of the underlying continuous nature of age.

6.1. Models

To predict age we consider three different models. These are

- Linear Least Squares: Least squares model where the mean squared error for predicted age is minimized with respect to the weights of the model.
- Gradient Boosting [2]: An ensemble method that iteratively fits weak learners to approximate the gradient of the loss. In a weighted sum these weak learners contribute to the overall result. Gradient boosting provides a general framework for fitting against almost arbitrary losses where AdaBoost [3] is limited.
- Fully Connected Neural Network: We use ReLU non linearity after each hidden layer. We experiment with different settings for the number of hidden layers and number of neurons in the hidden layers.

6.2. Predictors

As predictors we start with all features. That means the *128 dimensional graph embedding* plus *all pre-selected features* from the Pokec profiles. As a further step, for the best model, we consider subsets of all features in section 6.6

6.3. Metrics

In this subsection we define the usual metrics like Mean Absolute Error and Root Mean Square Error used for regression tasks. We also define our own metrics that give more insight into the perfromance of regression models for our specific task. Let y_i be the true age in years of sample i and $\hat{y}_i$ be the predicted age from the model in years. For evaluation we use the following metrics:

- Mean Absolute Error (MAE): Gives by how many years our prediction is off the true age (in years) on average. It is defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{1}$$

- Root Mean Squared Error (RMSE): is defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i)^2} \tag{2}$$

- Mean per Age Mean Absolute Error (MPA_MAE): Let S be the set of all age values in years with non zero number of samples. Let P_j be the set of samples of true age j. Then we define the MPA_MAE metric as:

$$\text{MPA\_MAE} = \frac{1}{|S|}\sum_{j\in S}\frac{1}{|P_j|}\sum_{y_i\in P_j}|\hat{y}_i - y_i| \tag{3}$$

- MPA_MAE p to q: This metric is similar to MPA_MAE but we consider only the range of ages $p, p+1, ...q$. Let S be the set of all age values from $p, p+1, ...q$ in years with non zero number of samples. Let P_j be the set of samples of true age j, then we define $\text{MPA\_MAE}_{p,q}$:

$$\text{MPA\_MAE}_{p,q} = \frac{1}{|S|}\sum_{j\in S}\frac{1}{|P_j|}\sum_{y_i\in P_j}|\hat{y}_i - y_i| \tag{4}$$

The MPA_MAE is used to address the imbalanced data set (age distribution shown in Figure1). The metric measures the performance independent of the sample count per age, because each MAE for a specific age contributes equally to the sum.

6.4. Training

We first split the dataset into two parts: 80% for training and 20% for testing. From the training set we we additionally set aside 10% as a validation dataset. We use this to choose the best hyperparameter setting for FCNN model and Gradient Boosting Model.

We perform grid search to select the best hyperparameters for the boosting model. For the FCNN we use the "baby sitting" approach. This means we manually search for hyperparameters based on reasonable arguments, observations while training and results.

6.4.1 Hyperparameters

The final hyperparameters we use are given in the Table 3 and 4.

Step size	0.1
Max tree depth	5
Iterations	400

Table 3: Hyperparameters for boosting

Hidden Layer Units	[256, 64, 32]
Activation Function	ReLU after each hidden layer
Learning Rate	0.0001
Weight Decay Rate	0.00001
Batch Size	2048
Epochs	750

Table 4: Hyperparameters for FCNN

6.5. Results

The results shown in Table 5 show that the FCNN gives the best MAE performance. On average the prediction is 2.62 years off of the true age. In comparison to the constant regressor this is an improvement of 61%. The FCNN is 11% better than the least squares regressor. Gradient Boosting gives the best result interms of the MPA_MAE metric. This means on average it performs better on a broader age range. This can also be seen in Figure 2 where we plot age vs MPA_MAE. In the range between 0 and 6 it performs better than the FCNN. In the range between 14 and 40 all regressors perform similarly with slightly better performance for the FCNN. This explains why FCNN performs best for the overall MAE metric as the data set is imbalanced towards this range.

Further investigating Figure 2, it seems that for ages higher than 65 all regressors perform very badly. But the reliability of the MPA_MAE metric is questionable here because of two reasons. One, the regressors have high variance in higher age ranges and little confidence because of the lack of training data in these ranges. Two, because of lack of test data, the MAE for higher ages is not meaningful. Since the range from 70 to 112 covers almost 40% of the full age range, the MPA_MAE metric suffers from high variance in our case and this makes it unreliable on the full age range. A regressor could just by chance outperform another in terms of the MPA_MAE in this age range. This is the reason why we consider a restricted version of the MPA_MAE, the $\text{MPA\_MAE}_{5,65}$ metric.

Overall the FCNN performs best on average on the imbalanced test set and in the range between 14 and 60. The boosting model performs better in the range between 0 and 6 and has a comparable performance to the FCNN in other areas, so it performs better in terms of the MPA_MAE metric especially in the range from 5 to 65. We consider the boosting model as the overall best model.

6.6. Feature Relevance

To measure the relevance of features for age prediction we

- Perform backward selection on the feature set for the boosting model. Features that are completely irrelevant for the prediction of age introduce only noise and by removing them the variance of the model decreases and generalization performance is improved. We aim on finding such features by performing a backward selection and assess generalization error.
- We consider different feature subsets. Even if a feature provides some useful information for predicting age, the importance can be negligible. So including the feature does not significantly improve the model performance. We aim on identifying a subset of especially important features.

For all following experiments we consider only the Gradient Boosting model as we consider this to be the best model based on the performance in all metrics.

6.6.1 Backward Selection

Only removing completion percentage leads to a slight improvement for the $\text{MPA\_MAE}_{5,65}$ metric on the test set, but worse performance for the MAE metric, as shown in Table 6. We do not consider the MPA_MAE over the whole age range further, because we do not consider it as reliable as described in section 6.5. Removing more features leads to worse results in comparison to using all features. So we stop backward selection at this point.

The result in this case does not fit the correlation results (see Table 2). Completion percentage does not have the smallest correlation among all features but removing it improves improves the performance for $\text{MPA\_MAE}_{5,65}$ metric. Another insight here is that removing weight or marital status leads to a more significant performance decrease than removing any other features. We conclude that none of the features is merely introducing noise in the context of age prediction and should be removed.

We now investigate if we can find features, such that removing them leads to minimal decrease in performance. We use the correlation results as a heuristic for relevance and show the performance of the model using only features with the strongest correlation (see 2).

Results in Table 7 show that removing all Pokec profile features except marital status and weight (8 features removed) leads to a performance decrease of only 4% for the MAE metric. Removing weight and marital status successively leads to a more significant performance decrease. Also using all features except weight or all features except marital status leads to the most significant performance decrease in comparison to removing other single features (see

	Constant	Least Squares	G-Boosting	FCNN
MAE	6.6398	3.3437	2.9925	2.6218
RMSE	8.8755	5.1658	4.8128	4.4634
MPA_MAE	29.1046	22.1357	20.8909	21.4934
MPA_MAE$_{5,65}$	16.9041	11.57696	7.1045	8.79366

Table 5: Results for age prediction using all pre-selected features

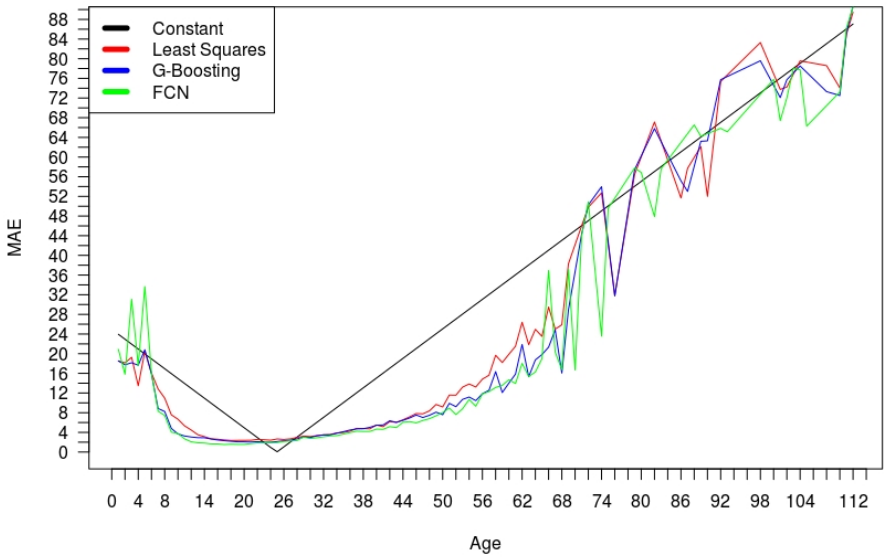

Figure 2: Per class age prediction using all pre-selected features on the test set.

Table 6). So on a coarser level correlation serves as a useful heuristic and we can conclude that weight and marital status are also from the prediction perspective the most important features from the Pokec profile data in combination with the graph embedding. Furthermore, we show that the graph embedding alone reaches better performance than the Pokec profile data.

7. Conclusions

Our best model predicts age with an error of 2.62 years on average for the imbalanced test set. In comparison to related work [8] we are able to improve the performance in terms of the Mean Absolute Error by 36% from 4.09 to 2.62. Since our dataset is imbalanced we introduce MPA_MAE$_{5,65}$ for which we get an error of 7.09 years. For feature relevance we show that the graph embedding provides the most useful information for age prediction but the model performance can be significantly improved with additional features. From these additional features we find that marital status and weight give the most significant improvement.

Given these result we conclude that it is possible to give a close estimate of the age of a user in an online social network. The user can share other information from which their age can be inferred. This raises a privacy issue. The attack is in principle possible in other OSNs like Facebook if the features that we used are available to the attacker. Relationships and marital status are basically part of the profiles in Facebook but need to be publicly shared. For defending against this attack especially sharing the friendship relationships and the marital status should be treated with care.

	RMSE	MAE	MPA_MAE	MPA_MAE 5 to 65
All	4.8128	2.9925	20.8909	7.1045
All/{public}	4.8187	3.0060	20.8106	7.1277
All/{completion percentage}	4.8229	3.0054	20.7719	7.0931
All/{gender}	4.8272	3.0134	20.7012	7.1337
All/{last login}	4.8280	3.0086	20.8776	7.1762
All/{registration}	4.8566	3.0414	20.8219	7.1712
All/{height}	4.8490	3.0188	20.9524	7.2421
All/{weight}	4.9443	3.0946	20.9528	7.4034
All/{comp education}	4.8471	3.0255	20.9238	7.1201
All/{smoking}	4.8256	3.0051	20.7669	7.1138
All/{marital}	5.1307	3.2253	21.6187	7.8782

Table 6: Different metrics on the test set after first step of the backward selection

	MAE	RMSE	MPA_MAE	$\text{MPA\_MAE}_{5,65}$
All	2.9925	4.8128	20.8909	7.1045
Graph emb + marital+ weight	3.1075	4.9428	22.4110	7.4658
Graph emb + marital	3.3587	5.2349	22.7625	7.9882
Graph emb	3.5987	5.6406	22.6643	9.1681
Pokec profile data	4.1096	6.0617	23.8951	9.4120

Table 7: Prediction performance on the test set for different features sub sets

References

[1] Pokec Social Network. https://pokec.azet.sk/.

[2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[3] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[5] Michal Zabovsky Lubos Takac. Data analysis in public social networks. https://snap.stanford.edu/data/soc-pokec.pdf, May 2012.

[6] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents*, SMUC '11, pages 37–44, New York, NY, USA, 2011. ACM.

[7] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA, 2014. ACM.

[8] Bryan Perozzi and Steven Skiena. Exact age prediction in social networks. In *Proceedings of the 24th International Conference on World Wide Web Companion*, WWW '15 Companion, pages 91–92, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[9] Jinxue Zhang, Xia Hu, Yanchao Zhang, and Huan Liu. Your age is no secret: Inferring microbloggers' ages via content and interaction analysis. In *Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016*, pages 476–485. AAAI Press, 2016.