Ranking social media news feeds: A comparative study of Personalized and Non-Personalized prediction models

Sami Belkacem[1], Kamel Boukhalfa[1], and Omar Boussaid[2]

[1] LSI laboratory, USTHB, Algiers, Algeria
{s.belkacem,kboukhalfa}@usthb.dz
[2] ERIC laboratory, University of Lyon 2, Lyon, France
omar.boussaid@univ-lyon2.fr

Abstract. Ranking news feed updates by relevance has been proposed to help social media users catch up with the content they may find interesting. For this matter, a single non-personalized model has been used to predict the relevance for all users. However, as user interests and preferences are different, we believe that using a personalized model for each user is crucial to refine the ranking. In this work, to predict the relevance of news feed updates and improve user experience, we use the *random forest* algorithm to train and introduce a personalized prediction model for each user. Then, we compare personalized and non-personalized models according to six criteria: (1) the overall prediction performance; (2) the amount of data in the training set; (3) the cold-start problem; (4) the incorporation of user preferences over time; (5) the model fine-tuning; and (6) the personalization of feature importance for users. Experimental results on *Twitter* show that a single non-personalized model for all users is easy to manage and fine-tune, is less likely to overfit, and it addresses the problem of cold-start and inactive users. On the other hand, the personalized models we introduce allow personalized feature importance, take into consideration the preferences of each user, and allow to track changes in user preferences over time. Furthermore, personalized models give a higher prediction accuracy than non-personalized models.

Keywords: Social media, News feed, Relevance, Personalization

1 Introduction

In several research approaches, ranking news feed updates in descending relevance order has been proposed to help social media users quickly catch up with the content they may find interesting in the news feed [1]. For this matter, supervised prediction models have been commonly used to predict the relevance of updates using labeled training data [2]. These models analyze past user behaviors to predict whether they will find an update relevant or not in the future [2]. However, in related work, to train a prediction model and predict the relevance, data of all users were first merged as if there is only one user. Then, a single non-personalized model has been trained on all data for all users. Indeed, according to Vougioukas et al. [1], in non-personalized models, a global model is typically trained on a large collection of updates received by multiple users and the interaction of each user with each update, e.g. retweets. The trained model is then used to predict a user-independent relevance score to each new update. By contrast, personalized models should be trained only on updates received by a particular user and the interactions of the particular user, e.g. whether the user retweeted each tweet. Hence, a separate model should be trained per user and then employed to provide user-specific relevance scores for each new tweet or, generally, social update. We believe that non-personalized models are useful to learn the overall interests of the majority of users (e.g., in general, users are likely to find relevant tweets that are similar to their own tweets), but generalize such unrealistic assumptions to all users makes it difficult to predict their individual preferences. For example, a given user might be more interested in new content that is different from his own tweets. Indeed, Paek et al. [3] noticed in their study 44 cases in which several participants had rated the same news feed post and found out that 82% of the cases differ in ratings suggesting that the relevance judgment can be subjective depending on the preferences of each user.

In this paper, we first provide background on ranking news feed updates according to a typical approach and a reminder of the non-personalized models used in related work. Then, to predict the relevance of news feed updates given that user preferences are different, we introduce a personalized prediction model for each user based on the *random forest* algorithm. Finally, we conduct a comparative study of personalized and non-personalized models according to six criteria: (1) the overall prediction performance of both approaches to get a global overview of the most effective model; (2) the amount of data in the training set to investigate the robustness of each model; (3) the cold-start problem, which is a common problem in recommender systems; (4) the incorporation of user preferences over time; (5) the model fine-tuning to investigate the manageability of each model; and (6) the personalization of feature importance for users.

The paper is structured as follows: section 2 presents background on ranking news feed updates on *Twitter*, section 3 provides a reminder of non-personalized prediction models, section 4 introduces our personalized model, section 5 discusses the experiments we performed to compare both models and highlight the need for personalization, and finally, section 6 concludes the paper.

2 Background

In this work, we focus like most related work on *Twitter*. Note, however, that it is possible to use this work on other social media platforms with some adaptations. Fig. 1 describes the primary non-personalized technique used to predict the relevance score $\boldsymbol{R(t,u)}$ of a tweet $\boldsymbol{t} \in \boldsymbol{F(u)}$, where $\boldsymbol{F(u)}$ denotes tweets unread by the recipient user $\boldsymbol{u}$ that can potentially be included in the news feed.

This technique is based on a supervised prediction model that analyzes labeled training data of tweets users read in the past to predict if the recipient user ***u*** will find the tweet ***t*** relevant in the future. Let ***D(u)*** denotes a subset of tweets previously read by the user ***u*** and ***D*** the overall labeled training data of all users. The training data of a user ***u*** is a set of input-out pairs such that an input represents a vector of features that may influence the relevance of a tweet ***t'*** ∈ ***D*** to ***u***, and the output represents the relevance score ***R(t',u)***. The primary technique involves three steps: (1) label tweets by relevance scores; (2) extract the features that may influence relevance; and (3) train the prediction model. In this section, we describe each step according to a typical approach [4].

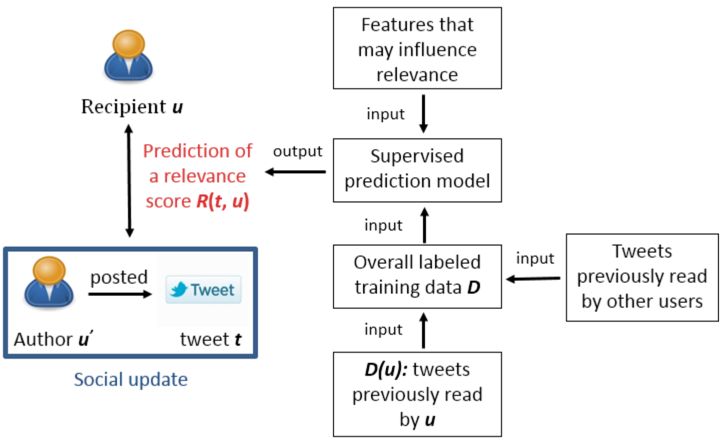

Fig. 1. Non-personalized prediction of a relevance score

First, to label tweets by relevance scores, we use the implicit method used by most related work [4]. It assumes that a previously read tweet ***t'*** ∈ ***D(u)*** is relevant to a user ***u*** if ***u*** interacted with ***t'*** (retweet, reply, like). Predicting relevance scores results in a binary classification problem. Note that some machine learning models such as *random forest* allow to predict the probability of classes and rank tweets by relevance according to the probability of having class 1.

Second, according to related work [4], we use 13 most relevant features that may influence the relevance of a tweet ***t***, posted by an author ***u'***, to the recipient ***u***. The features are divided in four categories, while more details are given in [4]:

- Features that match between the content of ***t*** and the interests of ***u***.
- Features that measure social tie strength between ***u*** and ***u'***. The assumption is that ***t*** could be relevant to ***u*** if ***u*** and ***u'*** are close friends.
- Features that measure the authority of ***u'***. The assumption is that ***t*** could be relevant to ***u*** if ***u'*** is important and has authority on the platform.

– Features that measure the quality of $\boldsymbol{t}$: length, popularity, if it has a photo, etc. The assumption is that $\boldsymbol{t}$ could be relevant to $\boldsymbol{u}$ if $\boldsymbol{t}$ is of high quality.

Finally, the prediction model aims to analyze labeled training data of tweets users read in the past to predict if they will find a tweet relevant in the future. Let $\boldsymbol{S}$ denotes the set of recipient users. First, we generate training data instances for each recipient user $\boldsymbol{u} \in \boldsymbol{S}$ in the form of input-output pairs considering each previously read tweet $\boldsymbol{t'} \in \boldsymbol{D(u)}$. An input represents a vector of features that may influence the relevance of $\boldsymbol{t'}$ to $\boldsymbol{u}$, and the output represents the implicit relevance score $\boldsymbol{R(t',u)}$. Then, we can either train a personalized prediction model for each user $\boldsymbol{u} \in \boldsymbol{S}$, or merge all data as if there was only one user to train a single non-personalized model for all users. The aim of both approaches is to map new input features of a tweet unread by a user $\boldsymbol{u}$ to a relevance score using a binary classifier learned from previously read tweets in the training set. In the next section, we provide a reminder of non-personalized models.

3 Non-personalized models

In non-personalized models, a single model is trained on a large collection of tweets received by multiple users and the interactions of all users with each tweet [1]. The trained model is then used to assign a user-independent relevance score to a new incoming tweet. Fig. 2 describes the primary technique used in related work to train a non-personalized prediction model. First, historical user data, which consists of previously read tweets $\boldsymbol{D}_i$, are merged and scaled to have feature values within the same range. Then, the overall data $\boldsymbol{D}$ is shuffled as if there is only one user and no chronological order of tweets. Finally, data is split into two sets: a training set to train the prediction model with 70% of the data and a test set to evaluate the performance with the 30% remaining data.

Table 1 indicates the non-personalized models used in related work. The table shows that different supervised algorithms were used: *logistic regression* [1, 5–7], *Support Vector Machines* [3], *artificial neural networks* [7–10], etc. In each work, a single algorithm was used for either: (1) all users [1, 8–13]; (2) each fold/partition of data with five folds in [3] and three partitions in [5]; or (3) each demographic subset of users [7]. In other words, no related work has used a single model for each user, such that in the best of cases, five models were used for 24 users in [3] and n models in [7], where n is the number of demographic subsets of users. The research work state that non-personalized models benefit from a large collection of tweets in the training set. Each tweet is represented as a feature vector that includes user-specific features. If two users receive the same tweet, it will be represented by two different feature vectors, which allows the model to produce different predictions per user for the same incoming tweet.

Nonetheless, since non-personalized models are trained on all data as if there is only one user, the models may learn and generalize unrealistic assumptions

Table 1. Non-personalized models in related work

Research work	Data	Supervised algorithm	A prediction model for
[11]	665 tweets	Coordinate ascent algorithm	All users
[12]	816 users	*Gradient Boosting*	
[13]	675 users	*Naive Bayes*	
[1]	122 users	*logistic regression*	
[10]	2 users	*artificial neural networks*	
[9]	307 users		
[8]	1000 users		
[3]	24 users	*Support Vector Machines*	Each fold of data (5 folds)
[5]	LinkedIn users	*logistic regression*	Each partition of data (3 partitions)
[6]			
Facebook [7]	Trillions of examples	*logistic regression, artificial neural networks, Gradient Boosting, etc.*	Each demographic subset of users

(e.g., all users are likely to find relevant the tweets that are similar to their own tweets). The importance/weight of the features learned by non-personalized models is assumed to be the same for all users, but such assumptions may not apply to some users. For example, a given user might be more interested in new content that is different from his own. Indeed, Paek et al. [3] asked 24 participants to rate news feed posts and noticed that 82% of ratings that concern the same tweets are different. This study indicates that the relevance judgment is subjective as user preferences and interests are different. Therefore, we believe that using a personalized user-dependent model is crucial to enhance the news feed content. In the next section, we introduce our approach that uses the *random forest* algorithm to train a personalized prediction model for each user.

4 A personalized prediction model

In contrast to non-personalized models, personalized models should be trained on tweets received by a particular user and the interactions of the particular user with each tweet. Hence, a separate model should be trained per user and then employed to provide user-specific relevance scores to new incoming tweets. Fig. 2 describes the technique we use to train a personalized prediction model for each user and assign user-specific relevance scores to tweets. First, we sort tweets by time and divide the training data $\boldsymbol{D}_i$ of each user $\boldsymbol{u}_i \in \boldsymbol{S}$ into two sets: a training set of the prediction model for the 70% least recent instances and a test set for the remaining 30% most recent instances. The purpose is to keep a chronological track of the relevance judgment of tweets by users over time. Then, we use the training set of each user $\boldsymbol{u}_i \in \boldsymbol{S}$ to train the corresponding *random forest* model $\boldsymbol{M}_i$. *Random Forest* [14] is a popular *ensemble learning* method[3] for classification and regression problems that operate by constructing a multitude of *decision trees*. In our previous work [2], we compared several machine learning algorithms used in related work and found that ensemble learning models are the most suitable to predict the relevance of news feed updates.

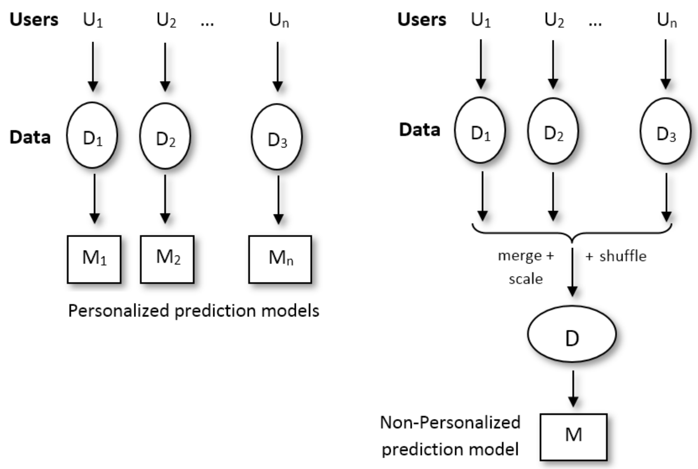

Fig. 2. Personalized and Non-Personalized models

The aim of using a personalized *random forest* model for each user is to make tailored recommendations, which may not coincide with the interests of the majority of users that non-personalized models are trained to predict. Indeed, unlike non-personalized models, not only the feature vector is different for each user-tweet pair, but also the feature importance/weight for each user. In other words, as a model is trained on the data of a given user independently of the other users, the model learns the individual user preferences and interests (e.g., a user interested in art is more likely to find tweets with a multimedia content relevant). Another reason to use a personalized model for each user is to sort and split the corresponding train and test data by time. Train the model on recent data allows to track changes in user preferences over time and make time-sensitive recommendations accordingly. In the next section, we describe the experiments we used to compare personalized and non-personalized models.

[3] A method that uses multiple machine learning algorithms to obtain better predictive performance than could be obtained by any of the constituent learning algorithms.

5 Experiments and comparison results

To compare personalized and non-personalized models and highlight the need for personalization, we describe in this section: (1) the dataset used in the experiments we performed; (2) the measures we used to evaluate the performance; (3) the methodology we used in the comparison; and (4) the obtained results.

5.1 Dataset

First, we randomly selected a set $\boldsymbol{S}$ of 46 recipient users and collected data over ten months using *Twitter Rest API*[4]. Then, to simulate the news feed of each user $\boldsymbol{u} \in \boldsymbol{S}$, we used the following principle to select, $\boldsymbol{D(u)}$, tweets posted by the followings of $\boldsymbol{u}$ that he may have read: (1) sort the tweets posted by the followings of $\boldsymbol{u}$ from least recent to most recent; (2) for each tweet $\boldsymbol{t'}$ with which $\boldsymbol{u}$ interacted, keep the chronological session defined by the tweet $\boldsymbol{t'}$, the tweet before $\boldsymbol{t'}$, and the tweet after $\boldsymbol{t'}$. This resulted in 26180 tweets, a 35% interaction rate with tweets and 569 tweets on average as training data for each user.

5.2 Measures

First, we train *random forest* classifiers for both personalized and non-personalized models using 70% of the data. Then, we define the following concepts to evaluate the models using the corresponding test set with 30% of the data [15]:

- True Positive (TP): # of relevant tweets correctly predicted relevant
- True Negative (TN): # of irrelevant tweets correctly predicted irrelevant
- False Positive (FP): # of irrelevant tweets incorrectly predicted relevant
- False Negative (FN): # of relevant tweets incorrectly predicted irrelevant

After that, we use the weighted *F1 score* measure given by Equation 1 [15], which is a popular measure for binary and unbalanced classification problems.

$$F = \frac{(F_r \times (TP + FN)) + (F_i \times (TN + FP))}{TP + TN + FP + FN} \tag{1}$$

Where:

- F_r is the standard *F1 score* for the class of relevant tweets
- F_i is the standard *F1 score* for the class of irrelevant tweets

5.3 Methodology

In the experiments, we first selected the best *random forest* parameters (number of trees, maximum three depth, splitting criterion, etc.) for a fair comparison between non-personalized and personalized models. Hence, a *random search* was run over different parameter values so that the parameters are optimized by

[4] https://dev.twitter.com/rest/public

a cross-validated search [16]. Indeed, we used a cross-validation for the non-personalized model and a time-series cross-validation for the personalized model as the latter preserves the chronological order of tweets [15], unlike the non-personalized model where data is shuffled. Then, to study the model stability with several runs and small changes to training data, we retrained each model on 30 different random state[5] values and evaluated it on the test set. Finally, we select the average *F score* for personalized and non-personalized approaches.

5.4 Results

The comparison and evaluation results are presented and discussed according to six criteria: (1) the overall prediction performance of both approaches to get a global overview of the most effective model; (2) the amount of data in the training set to investigate the robustness of each model; (3) the cold-start problem, which is a common problem in recommender systems; (4) the incorporation of user preferences over time; (5) the model fine-tuning to investigate the manageability of each model; and (6) the personalization of feature importance for users.

First, the results show that introducing a personalized model for each user has improved the average *F score* by +3.12%, from 77.73% with the non-personalized model to 80.85% with the personalized model. Therefore, to make refined predictions and select the tweets that might be relevant to a given user, it is more convenient to train a model on tweets the user has found relevant in the past rather than including tweets and behaviors about other users in the training process. Undoubtedly, tweets that are relevant to one user are not necessarily relevant to another user, which illustrates the importance of the personalized model we introduce to capture individual user needs and improve the prediction accuracy. Time-aware user preferences are another advantage of personalized models that makes them more accurate. Indeed, train the model on recent data allows time-sensitive recommendations. The personalized models capture the chronological evolution of user relevance judgment of tweets, which may change with time (e.g., a user may over time give less importance to popular tweets and more importance to tweets related to his interests). In contrast, the non-personalized model cannot predict such behaviors since data of all users are merged and shuffled as if there is only one user and no chronological order of tweets.

Second, we computed feature importance values[6] [14] in both personalized and non-personalized models, which are presented in Table 2 and Fig. 3 respectively. Fig. 3 gives the average feature importance for all users. The figure shows that non-personalized models can learn and provide an overview of the features that influence the relevance judgment of tweets by users, which is useful

[5] A variable used in randomized machine learning algorithms to determine the random seed of the pseudo-random number generator

[6] *Random Forest* computes the importance of a feature as the normalized total reduction of the criterion brought by that feature, also known as the *Gini importance*.

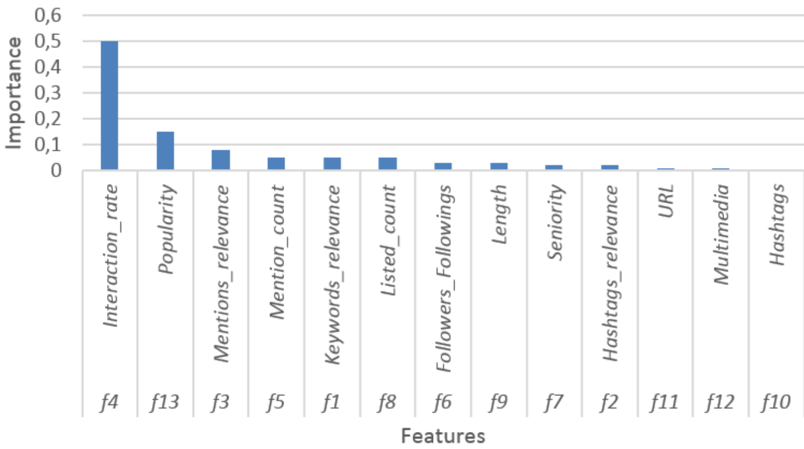

Fig. 3. Non-Personalized feature importance

to understand user behaviors and the assessment of relevance in general. For example, the top feature is the feature f_4 (0.5), the interaction rate of ***u*** with tweets posted by ***u'***. In contrast to non-personalized feature importance, Table 2 gives the personalized feature importance for each recipient user. First, we note that feature importance differs according to users, i.e. features that are important to one user are not necessarily important to another user, e.g. the feature f_9 which stands for the tweet length is very important to the user *Red_ or_ MC1R* when judging the relevance of tweets (0.22) but not to the user *Medium* (0.02). Certainly, user preferences are different, and this illustrates the gain brought by a personalized prediction model for each user, which takes into consideration individual interests. Furthermore, we note that the features learned as highly important by the non-personalized model are in fact, not important to all users. For example, the feature f_4 (0.5), the interaction rate of ***u*** with tweets posted by ***u'***, is important to many users when judging the relevance of tweets, but not to some users, e.g. the users *TheMuslimReform* (0.01), *LKrauss1* (0.02), and *bamwxcom* (0.03). This proves that non-personalized models generalize unrealistic assumptions to all users. In opposite, personalized models allow tailored recommendations that are different from the preferences of the majority of users.

Despite all the improvements the personalized models have brought in, we observe from the evaluation results that the proposed approach has some limitations. Fig. 4 presents the learning curve of the non-personalized model for all users along with the learning curve of the personalized model for the user *ch402*. Note that the learning curves of the 46 users in the dataset are quite similar; hence we randomly selected a single user as a case study due to lack of space.

First, Fig. 4 shows that the non-personalized model benefits from a large col-

Table 2. Personalized feature importance

User	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13
Astro0Glen	0.02	0.04	0.0	0.39	0.06	0.13	0.07	0.1	0.04	0.01	0.01	0.0	0.14
Astro_Pam	0.05	0.06	0.03	0.23	0.06	0.09	0.05	0.1	0.09	0.01	0.03	0.02	0.2
bamwxcom	0.0	0.01	0.13	0.03	0.0	0.1	0.06	0.08	0.08	0.01	0.02	0.08	0.39
Baronatrix	0.03	0.01	0.0	0.21	0.0	0.11	0.09	0.15	0.12	0.01	0.04	0.01	0.23
BethStamper6	0.04	0.0	0.0	0.3	0.0	0.06	0.06	0.1	0.14	0.02	0.04	0.02	0.22
byudkowsky	0.03	0.0	0.0	0.14	0.12	0.21	0.05	0.12	0.17	0.0	0.01	0.0	0.14
ch402	0.03	0.0	0.01	0.19	0.02	0.11	0.07	0.15	0.13	0.01	0.02	0.01	0.25
demishassabis	0.06	0.0	0.02	0.25	0.0	0.09	0.23	0.09	0.08	0.01	0.01	0.0	0.17
eevil_abby	0.2	0.0	0.07	0.06	0.05	0.06	0.05	0.09	0.19	0.02	0.01	0.03	0.18
elonmusk	0.04	0.0	0.0	0.23	0.02	0.1	0.06	0.07	0.06	0.01	0.02	0.01	0.39
GeorgeHarrison	0.0	0.01	0.0	0.2	0.22	0.07	0.03	0.15	0.02	0.0	0.0	0.0	0.29
GilmoreGuysShow	0.03	0.0	0.0	0.1	0.0	0.14	0.08	0.19	0.13	0.02	0.07	0.01	0.22
gwern	0.03	0.0	0.0	0.21	0.01	0.16	0.07	0.11	0.08	0.04	0.01	0.0	0.29
homebrew	0.02	0.01	0.01	0.2	0.09	0.07	0.03	0.1	0.08	0.01	0.01	0.01	0.37
HybridZizi	0.04	0.0	0.0	0.3	0.0	0.11	0.06	0.13	0.1	0.02	0.03	0.02	0.21
jadelgador	0.08	0.01	0.0	0.29	0.04	0.07	0.05	0.09	0.03	0.01	0.0	0.01	0.33
JHUBME	0.05	0.04	0.11	0.17	0.13	0.08	0.04	0.11	0.07	0.01	0.0	0.0	0.19
JohnDawsonFox26	0.04	0.02	0.0	0.26	0.12	0.07	0.04	0.06	0.05	0.01	0.02	0.08	0.25
john_walsh	0.0	0.21	0.02	0.13	0.03	0.14	0.03	0.17	0.05	0.03	0.0	0.0	0.18
kilcherfrontier	0.01	0.12	0.12	0.27	0.07	0.07	0.04	0.19	0.03	0.0	0.02	0.0	0.06
LKrauss1	0.05	0.0	0.08	0.02	0.06	0.11	0.06	0.15	0.17	0.04	0.14	0.01	0.12
mastenspace	0.14	0.0	0.3	0.23	0.01	0.05	0.02	0.17	0.02	0.0	0.0	0.0	0.05
Medium	0.4	0.0	0.21	0.06	0.0	0.03	0.01	0.16	0.02	0.02	0.0	0.0	0.09
microphilosophy	0.13	0.01	0.0	0.11	0.01	0.12	0.16	0.08	0.1	0.02	0.01	0.01	0.25
MIRIBerkeley	0.23	0.0	0.0	0.09	0.04	0.09	0.18	0.12	0.08	0.01	0.01	0.0	0.15
NASAKepler	0.12	0.07	0.31	0.03	0.06	0.04	0.01	0.15	0.1	0.0	0.0	0.0	0.1
NASA_Wallops	0.08	0.05	0.03	0.11	0.02	0.07	0.0	0.18	0.03	0.0	0.05	0.0	0.38
newscientist	0.26	0.0	0.19	0.07	0.0	0.08	0.03	0.1	0.07	0.01	0.0	0.02	0.18
PattiPiatt	0.03	0.01	0.0	0.34	0.01	0.07	0.11	0.12	0.06	0.02	0.06	0.01	0.17
peterboghossian	0.05	0.01	0.06	0.23	0.03	0.08	0.05	0.13	0.1	0.01	0.02	0.01	0.22
rafat	0.03	0.0	0.03	0.18	0.0	0.11	0.04	0.11	0.08	0.01	0.03	0.04	0.33
realDonaldTrump	0.02	0.03	0.0	0.1	0.0	0.12	0.02	0.07	0.03	0.0	0.0	0.01	0.58
Red_or_MC1R	0.03	0.0	0.0	0.11	0.0	0.13	0.06	0.13	0.22	0.02	0.05	0.01	0.24
renormalized	0.03	0.0	0.0	0.16	0.0	0.11	0.07	0.13	0.13	0.0	0.06	0.01	0.3
RossTuckerNFL	0.03	0.0	0.22	0.2	0.05	0.09	0.09	0.13	0.04	0.01	0.02	0.02	0.11
RoxanneDawn	0.02	0.04	0.0	0.28	0.03	0.14	0.07	0.16	0.07	0.01	0.02	0.02	0.15
scimichael	0.06	0.03	0.0	0.32	0.0	0.14	0.04	0.12	0.05	0.01	0.0	0.0	0.24
SfNtweets	0.04	0.18	0.02	0.13	0.02	0.06	0.13	0.12	0.08	0.01	0.01	0.0	0.2
slatestarcodex	0.01	0.0	0.0	0.14	0.0	0.22	0.09	0.13	0.13	0.0	0.03	0.0	0.25
SLSingh	0.06	0.0	0.0	0.23	0.02	0.11	0.11	0.2	0.03	0.0	0.02	0.0	0.21
sxbegle	0.04	0.0	0.14	0.17	0.02	0.09	0.05	0.11	0.1	0.01	0.02	0.0	0.23
TeslaRoadTrip	0.05	0.01	0.0	0.23	0.0	0.12	0.06	0.1	0.04	0.01	0.02	0.03	0.34
TheMuslimReform	0.04	0.0	0.0	0.01	0.24	0.13	0.13	0.1	0.11	0.02	0.0	0.0	0.22
TheRickDore	0.01	0.02	0.22	0.1	0.0	0.12	0.03	0.11	0.08	0.02	0.05	0.02	0.2
USDISA	0.07	0.01	0.12	0.16	0.03	0.12	0.04	0.13	0.12	0.02	0.01	0.01	0.16
WestWingWeekly	0.04	0.02	0.14	0.21	0.11	0.09	0.04	0.07	0.08	0.01	0.01	0.01	0.18

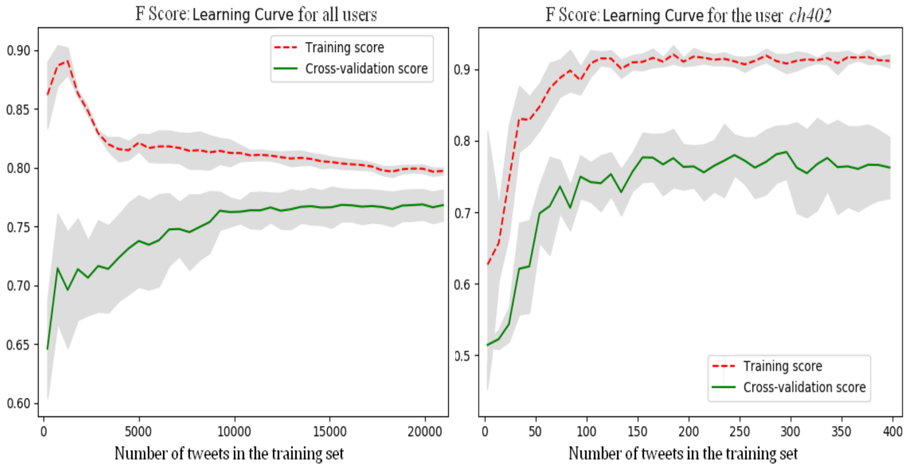

Fig. 4. Learning curves: Non-Personalized (left) vs. Personalized (right)

lection of tweets in the training set compared to the personalized model (20000 against 400 tweets). Indeed, unlike personalized models which are trained on the individual data of each user, the non-personalized model merges the data of all users, which allows it to be trained on a large collection of tweets received by different users. Note that in the dataset, there is an average of 569 tweets in the training database of each recipient user and a median of 343 tweets.

Second, the training and cross-validation curves in Fig. 4 indicate that both personalized and non-personalized models converge, suggesting that the models are able to learn to classify tweets according to their relevance. However, we notice that train a non-personalized model on a larger training set makes it more robust and less likely to overfit comparing to the personalized model. In other words, the training and cross-validation curves of the non-personalized model converge to the same *F score* value (76%), indicating that the model can generalize relevance predictions to unseen tweets. As to the personalized model, which is trained on a smaller training set, we observe that the model fits the training dataset too well with a high *F score* value (90%) and loses some of its ability to generalize to the cross-validation set with a lower *F score* value (78%). Therefore, to make more accurate predictions to new and unseen tweets, it would be advisable to use one of the many machine learning techniques to prevent overfitting: regularization, early stopping, data augmentation, etc. [15].

Finally, another notable difference is that non-personalized models may work better with new or inactive users, for which personalized models may have very few training instances. Indeed, in such cases, the personalized model does not have information about user preferences and interests in order to make specific recommendations. Hence, it is important to suggest alternatives to address this common problem in recommender systems known as the *cold-start problem*. Non-personalized models address this issue by default since the same model can be used to any user on the social media, even new or inactive users. Lastly, it is easier for social media administrators/developers to fine-tune and manage a single non-personalized model than fine-tuning a personalized model for each user. For example, in our case, it was somewhat possible to look at each of the 46 prediction models corresponding to the 46 recipient users, but this may become more challenging as the number of users increases. In such a situation, it is necessary to provide reliable automatic techniques to validate user models.

6 Conclusion

In this paper, to predict the relevance of news feed updates and improve user experience, we used the *random forest* algorithm to train and introduce a personalized prediction model for each user. Then, we conducted a comprative study of personalized and non-personalized models according to six criteria: (1) the overall prediction performance; (2) the amount of data in the training set; (3) the cold-start problem; (4) the incorporation of user preferences over time; (5) the model fine-tuning; and (6) the personalization of feature importance for users. The experimental results on *Twitter* show that a single non-personalized model for all users is easy to manage and fine-tune, is less likely to overfit as it benefits from more data, and it addresses the problem of cold-start and inactive users. On the other hand, the personalized models we introduce allow personalized feature importance, take into consideration the preferences of each user, and allow to track changes in user preferences over time. Furthermore, the personalized models give a higher prediction accuracy than non-personalized models. These findings highlight the need for personalization to effectively rank the news feed.

Despite the advantages that personalized models have brought over the classical non-personalized models, we observed that non-personalized models may still work better with new or inactive users, for which personalized models may have very few training instances. Hence, it is important to suggest alternatives to address this common problem in recommender systems known as the *cold-start problem*. Non-personalized models address this issue by default since the same model can be used for any user, even new or inactive users. To address this problem, for example, it would be interesting to propose a hybrid method that takes the advantages of both personalized and non-personalized models.

References

1. Michail Vougioukas, Ion Androutsopoulos, and Georgios Paliouras. Identifying Retweetable Tweets with a Personalized Global Classifier. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence - SETN '18*, pages 1–8, Patras, Greece, 2018. ACM Press.

2. Sami Belkacem, Omar Boussaid, and Kamel Boukhalfa. Ranking news feed updates on social media: A comparative study of supervised models. In *EGC - Conference on Knowledge Extraction and Management*, volume 36, pages 499–506. Revue des Nouvelles Technologies de l'Information, 2020.
3. Tim Paek, Michael Gamon, Scott Counts, David Maxwell Chickering, and Aman Dhesi. Predicting the Importance of Newsfeed Posts and Social Network Friends. In *AAAI*, volume 10, pages 1419–1424, 2010.
4. Sami Belkacem, Kamel Boukhalfa, and Omar Boussaid. Expertise-aware news feed updates recommendation: a random forest approach. *Cluster Computing*, 23(3):2375–2388, September 2020.
5. Deepak Agarwal, Bee-Chung Chen, Rupesh Gupta, Joshua Hartman, Qi He, Anand Iyer, Sumanth Kolar, Yiming Ma, Pannagadatta Shivaswamy, and Ajit Singh. Activity ranking in LinkedIn feed. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1603–1612, 2014.
6. Deepak Agarwal, Bee-Chung Chen, Qi He, Zhenhao Hua, Guy Lebanon, Yiming Ma, Pannagadatta Shivaswamy, Hsiao-Ping Tseng, Jaewon Yang, and Liang Zhang. Personalizing linkedin feed. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1651–1660, 2015.
7. Lars Backstrom. Serving a Billion Personalized News Feeds. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining - WSDM '16*, pages 469–469, San Francisco, California, USA, 2016. ACM Press.
8. Qi Zhang, Yeyun Gong, Jindou Wu, Haoran Huang, and Xuanjing Huang. Retweet Prediction with Attention-based Deep Neural Network. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 75–84, Indianapolis Indiana USA, October 2016. ACM.
9. Guangyuan Piao and John G. Breslin. Learning to Rank Tweets with Author-Based Long Short-Term Memory Networks. In *International Conference on Web Engineering*, pages 288–295. Springer, 2018.
10. Carmen De Maio, Giuseppe Fenza, Mariacristina Gallo, Vincenzo Loia, and Mimmo Parente. Time-aware adaptive tweets ranking through deep learning. *Future Generation Computer Systems*, 93:924–932, April 2019.
11. Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, page 2261, Glasgow, Scotland, UK, 2011. ACM Press.
12. Keyi Shen, Jianmin Wu, Ya Zhang, Yiping Han, Xiaokang Yang, Li Song, and Xiao Gu. Reorder user's tweets. *ACM Transactions on Intelligent Systems and Technology*, 4(1):1–17, January 2013.
13. Kaisong Song, Daling Wang, Shi Feng, Yifei Zhang, Wen Qu, and Ge Yu. CTROF: A Collaborative Tweet Ranking Framework for Online Personalized Recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 1–12. Springer, 2014.
14. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
15. Claude Sammut and Geoffrey I. Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.
16. James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305, 2012.