Understanding Email Interactivity and Predicting User Response to Email

Soumyadeep Roy, Nibir Pal, Kousik Dasgupta and Binay Gupta

1 Introduction

Organizations face difficulty in addressing email overload due to the growing customer base and the increased volume of communications both within the organization and outside. Understanding how users interact with email helps organizations to develop targeted strategies by understanding the customer base and also helps in improving the organizational productivity. Controlling email overload at work either by adopting software that is designed for making email easier to use or by adopting effective tactics for using email as a communication medium significantly improves the coordination at work [5]. The average time taken by the employees to recover from an email interrupt and return to the same work rate at which they left it is 64 seconds [2]. As desktop search, machine learning and text processing techniques improve, and we utilize them in order to tackle the problems in task management [11] and in email overload. Intelligent agent-based systems and different information extraction techniques are used to extract content information from email and also generate meaningful user summaries based on the task at hand.

Most of the existing studies have worked on small samples of data or have performed surveys on employees of a given organization. The collected feedback contain details regarding recipient actions like delete, leave in inbox, file, reply, or plan immediately or plan to reply later. Some studies [4, 12] have large data samples. The work in [10] is extended by [3] where they perform analysis on intelligently categorizing the messages and determining its importance to the user. Previous research studies

S. Roy (✉)
Indian Institute of Technology Kharagpur, Kharagpur 721302, India
e-mail: soumyadeep.roy9@iitkgp.ac.in

N. Pal
Dynamic Digital Technology, Kolkata 700091, India

K. Dasgupta
Kalyani Government Engineering College, Kalyani, Nadia 741235, India

B. Gupta
dishq, Bengaluru 560102, India

J. K. Mandal et al. (eds.), *Methodologies and Application Issues of Contemporary Computing Framework*, https://doi.org/10.1007/978-981-13-2345-4_6

typically involve small user samples and do not provide insightful models which will help to better understand the recipient action with email [3].

Email rhythms are understood more through relationships than through isolated messages. Email expectation is influenced by the recipient, urgency of topic conveyed by using markers in the subject field or priority flag and also combining it with voice mail and based on the time shift between the sender and the recipient [10]. Emails are also varied based on the content and the recipient type. The expectation and breakdown points are not the same for every email, or even for every recipient. Mackay categorizes the recipients into two categories—prioritizers, who actively maintain good control over their inbox and address the messages as they arrive and archivers, who preserve the email messages for future use and actively ensure that no important messages are missed [9]. The current backlog of emails of a recipient is a significant factor since if one has unusual number of unread messages, one may not be able to maintain the usual responsiveness [10]. This formed the second focus of our study, for which we took into consideration the user preferences extracted from its own email communication history. Certain message content particularly of task management and delegation, scheduling, information exchange, and social communication also changes the importance of a message. A significant proportion of email does not even require any response since many messages are broadcasted to users for informational purposes or send to an additional recipient who is not the primary recipient. The messages that require responses from the recipients are the ones that require user attention and cognitive effort [10]. Sender characteristics accounted for an additional 15% of the total variance in the likelihood of responding to an email [3].

In this paper, we study email interactivity patterns on a very recent dataset from the year 2016 and we observe certain similar patterns to [7] and [6]. Throughout the paper, we use the term mail and email interchangeably implying the same meaning. We determine the feature importance based on information gain. We then use a novel methodology of developing user and send day profiles using k-means clustering as a feature space reduction step, since our training dataset is very large. We then use these features to train three distinct binary classifiers using Logistic Regression, Naive Bayes, and Decision Tree, to predict an email recipient response. Specifically, we predict whether the email recipient will open the email or not.

The paper is organized as follows. In Sect. 2, we discuss the related works in this domain. Section 3 presents the dataset details. In Sect. 4, we derive the feature importance based on information gain and explain how the final feature set is prepared after undergoing the stages of feature engineering and dataset profiling. In Sect. 5, we provide the results obtained by understanding the interactivity of emails with respect to the profiling of the dataset using k-means clustering. We also explain how the features are derived with respect to our classification models. We then compare the performance of our classifiers based on Logistic Regression, Decision Tree, and Naive Bayes. In Sect. 2, we provide the important results.

2 Related Works

Survey information is used to predict recipient action on specific messages as a function of message characteristics like message importance and relationship to the sender and content [3]. Interviews are also a good way to elicit recipient perceptions on email usage but they are unable to verify to what extent these perceptions are actually substantiated in reality at work [10]. Existing works use machine learning and information extraction techniques for addressing email overload. Logs of how quickly people respond to particular email senders over time can be analyzed to develop a response time prediction for a message. We can also determine a threshold to estimate the time instance after which breakdown has probably occurred.

One study [4] use both local(individual user-specific) and global(sender-specific) features to predict the actions—read, reply, delete, and delete-without-read, instead of determining the importance of the email to the user. In another study [12], both dyadic(one-to-one) and one-to-many email communication in an enterprise email setting is considered. They consider factors like email content and metadata, historical interactions, and temporal features for characterizing the email reply behavior prediction. A large-scale log analysis is carried out in [1] in order to understand differences in query formulation, refinding patterns and intent of search between email search and web search. Another study [7] analyzes the factors like stage as well as history of conversation, email load, day of week, time message is received, user demographics and use of portable devices, for determining the time taken to reply and length of reply. Contrary to previous works, they focus on quantitative measures of overload and its corresponding effect on the user behavior. A study [8] explores the impact of the frequency of checking emails on subjective well-being.

3 Dataset

Our dataset contains metadata of emails sent from HackerRank to the users who have opened their profiles in the HackerRank platform. They are referred to as hacker in the dataset documentation. It is taken from a HackerRank contest[1] held from August 29 to September 4, 2016. It has missing values in hacker timezone and mail category and has inconsistent values in hacker created at, last online, opened, clicked, unsubscribed. Both continuous and categorical attributes are present in the dataset. The training dataset consists of 400 k email records across 54 features, which are sent between February 12 and May 7, 2016. The test dataset cover emails sent between May 8 and May 17, 2016, totaling to 86,048 email records across 48 features. We remove the user reaction information, mentioned in Sect. 3.3 during the testing phase.

[1] https://www.hackerrank.com/machine-learning-codesprint.

3.1 Contest Information

The features are contest login count, contest participation count, submission count, and contest submission count and their values are provided in terms of last 1, 7, 30, 365 days and from the time of joining HackerRank.

3.2 Account Information

The features are hacker created at, forum comments count, forum count, forum expert count, forum question count, hacker confirmation, user id, hacker timezone, in-product notifications (ipn) count, ipn read, last online, mail category, mail id, mail type, and sent time. These features provide details about the recipient and also include its participation history details in the HackerRank community.

3.3 User Reaction Information

The features are click time, clicked, open time, opened, unsubscribe time, and unsubscribed. These features are associated with the user reactions. They do not appear in the test dataset.

4 Predicting User Response to Email

In this section, we first preprocess and normalize the dataset separately for the different binary classification models. We aggregate the user metadata into user profiles and the sent mail metadata into sent day based profiles by using k-means clustering. We perform feature engineering in stages, where first we derive the feature importance and then reduce the feature set from 62 features to 6 features, through dataset profiling, which we will discuss in detail later in this section. The derived set of features is used for training the binary classification models, Logistic Regression, Decision Tree, and Naive Bayes. Then, we compare the performance of proposed binary classification models and evaluate it with accuracy, precision, recall, and F1 score.

4.1 Preprocessing

We first clean the dataset by removing the observations with missing mail category values and logically inconsistent data conditioned as: hacker created at > last online; opened = false and clicked = true; opened = false and unsubscribed = true. The missing values of hacker timezone is filled with its median value. Since all the observations start on and after February 12, 2016, we perform normalization on sent time, hacker created at and last online, by subtracting the minimum value of the epoch from the time-stamped attributes. This modification does not affect the relative temporal ordering. In order to address the multicollinearity problem, we compute pairwise feature correlation values. For each feature pair having high correlation values between 0.8 to 1.0 and −0.8 to −1.0, we only keep only one feature out of them. The mail id and user id features are removed. Decision Tree and Naive Bayes classifier can work with categorical variable. In order to work with Logistic Regression, the categorical variables need to be first converted to binary-valued vectors using one-hot encoding.

We have data from the different timezones and it causes difficulty in understanding the email usage pattern in terms of hour of the day, when mails are more frequently sent and opened, corresponding to each timezone separately, as shown in Fig. 1. For the purpose of consistency, we only consider the data points belonging to the timezone 18,000, which has the highest number of observations among all the other timezones.

4.2 Dataset Profiling

After the initial cleaning of the dataset, in this part, we first perform feature engineering, where we derive new features like total mid open, total high open, max open time gap. We then perform feature selection, where we use k-means clustering to perform dataset profiling. This step results in our feature set to be reduced to only six features.

4.2.1 User Profile

Each email recipient identifier is mapped to its user number. We have a total of 28,509 unique users across the entire dataset. More than one data point may be associated with a single email recipient. We assign each user in the user profile dataset to their most recent profile data. We assign the cluster labels obtained after k-means clustering based on user characteristics, as a feature named user cluster number. We now have 73 attributes. We obtain the optimal set of parameter values, after trying different combinations of: k (number of clusters), nstart (the number of random initial cluster sets) and iter.max (maximum number of iterations allowed).

4.2.2 Sent Day Profile

We perform k-means clustering over the sent day profile dataset. We add the features like age of recipient in weeks and percentage of mails opened among all the mails

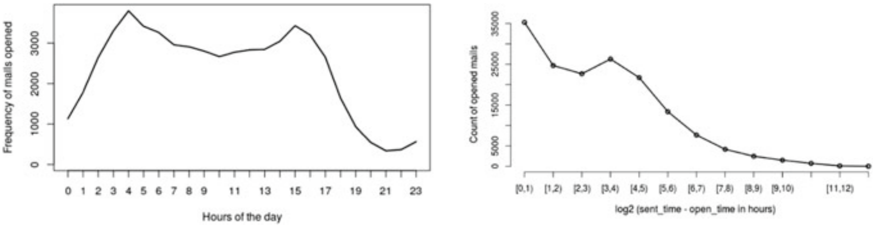

Fig. 1 (left) Variation of mails opened count over each hour of the day in timezone 18,000. (right) Mails opened count distribution against time gap between mail sent and opened in $\log_2$hours

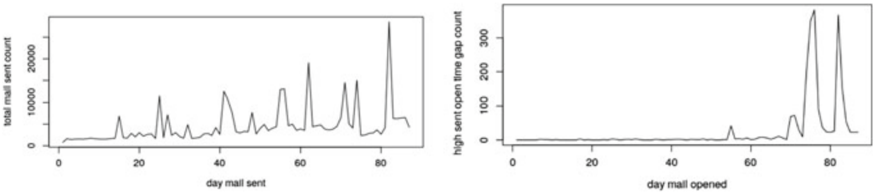

Fig. 2 (left) Variation of total mail sent by HackerRank over all 87 sent days. (right) Variation of high sent open time gap count over all 87 sent days

that are sent on that day and mails that are directly open that day and their temporal behavior is shown in Fig. 2. We also use the user activity related to clicked and unsubscribed actions on emails, as features. We add new features like total mid open, total high open, and max open time gap and analyze their effect on each other. We further preprocess them using min-max normalization.

4.3 Feature Importance

Here, we determine the variable importance of the derived feature set, which is then used for performing feature selection. The features are ranked based on the information gain associated with a Decision Tree classifier. Given the current feature set, we observe that the decision tree classifier cannot linearly split the data, when trying to predict the opened attribute. Upon further inspection based on the top five features based on higher information gain, we cannot differentiate between the mails that are opened and those not opened. We observe similar value distribution as shown in the corresponding box plots in Figs. 3 and 4.

We then focus on dataset profiling in terms of the recipient type and the day it is sent. This significantly improve the model performance and may be because we are using the recipient response history. This characteristic is captured by the opened percent attribute, which also shows a very high variable importance in the classification models developed.

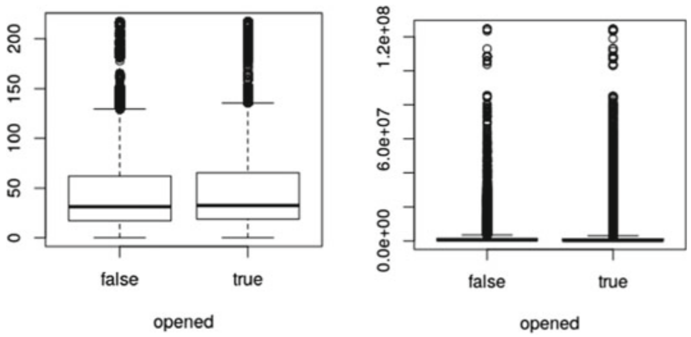

Fig. 3 (left) User profile age in weeks. (right) Difference between sent time and last online diff

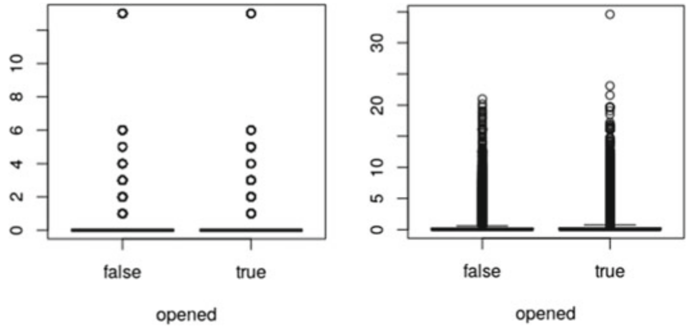

Fig. 4 (left) Forum expert count. (right) Ipn read aggregate

4.4 User Response Prediction

Importance of a message plays a significant role in users action on a message [3] and deduces that the mail content, sender characteristics, and user details that are available from the dataset are crucial for deriving new insights and usage patterns. Since it is proving difficult to construct one-model-for-all approach, we approach it by developing personalized models, which we capture in this study through dataset profiling. After dataset profiling, we finally have the final training dataset, which is now used for predicting the response of an email recipient. We now compare the performance of three distinct binary classification models—Logistic Regression, Decision Tree, and Naive Bayes, over the evaluation metrics namely, accuracy, precision, recall, and F1 score.

4.4.1 Logistic Regression

Logistic regression cannot directly work with categorical variables. In our study, where no ordinal relationship exists in the categorical variables, we use one-hot encoding to convert the categorical data to numerical data as a preprocessing step. The training set now contain 83 features. The glm function in R is used to train with the arguments: $family = binomial$. We use the predict function for testing purposes and use the arguments: $type = response$. This indicates that the output will be in the form of predicted probabilities. We then determine an optimal threshold to assign a predicted class, given a prediction probability.

4.4.2 Decision Tree

Decision Tree can handle both categorical as well as continuous attributes and therefore the dataset require no further transformation. The training dataset now has 15 features. The model is trained using the rpart R package with default arguments. The test output is predicted using predict. glm function with the arguments: type $= class$

4.4.3 Naive Bayes

We require the same preprocessing as required for the Decision Tree classifier, since Naive Bayes classifier can also handle categorical and continuous features. The training dataset has six features. We use the naiveBayes R package of e1071 library to train the model using the argument: type = class. The test output is generated using its predict function.

5 Results

We observe comparable patterns with the existing works from the email interactivity experiments. We then experimentally derive that the dataset for the task of user response prediction is not linearly separable, given the current feature set. We observe it by comparing the value distribution among the different features for each recipient response status. We then address this problem by performing user profiling along with sent day profiling. We can now train three binary classification models using Logistic regression, Decision Tree and Naive Bayes and achieve good performance for the email recipient response prediction task.

5.1 Email Interactivity

We observe a trend in the recipient and sender interaction, in terms of the volume over days, time of day and day of the week, versus the number of mails sent from HackerRank that are opened by the recipients in Fig. 2. The plots obtained give the intuition to develop separate profiles in terms of recipient preferences (uniquely identified by user id) and sent day (derived from sent time), in terms of sudden spikes and behavior specific to the email dataset and in terms of the quantity of mails sent and distinct behavior among the users as shown in Fig. 1. The mail records in the dataset are opened over a duration of 149 days. We also observe certain sudden spikes of high magnitude. We find an inconsistent trend in terms of percent of positive user response over the days the mails were opened, when plotted against the number of mails that crossed a high threshold of time gap between mail sent and when it was opened. The overall email interactivity in terms of the day of the week, time gap between mails sent and opened, opened and clicked, opened, and unsubscribed is also studied.

5.2 Dataset Profiling

We perform k-means clustering over the user profile dataset with argument: k = 55 and nstart = 20, iter.max = 50. The optimum value of (Between_SS/Total_SS) is 0.892.

After performing k-means clustering over the sent day profile dataset, with argument: k = 5 and nstart = 20. The optimum value of (Between_SS/Total_SS) is 0.853. We try to visualize the clusters using the pair of features having high variance with respect to the sent day profile dataset, which are: gap norm and mid open norm.

5.3 User Response Prediction Comparison

After performing feature selection, we have six attributes and this is the final dataset on which we will apply the three binary classification algorithms, Logistic Regression, Decision Tree, and Naive Bayes. We find that the opened percent attribute has a very high variable importance, which masks the effect of a significant feature—profile age, which is calculated in terms of the number of weeks covered starting from when the hacker profile is created. The features in accordance to their decreasing order of importance are: open percent, user cluster number, sent time gap, hacker timezone, hacker confirm, and profile age in weeks.

We compare the performance of three binary classification models of Logistic Regression, Decision Tree, and Naive Bayes in Table 1. For the current dataset, I

Table 1 Comparison of performance for different models

Approach	Accuracy	Precision	Recall	F1
Logistic regression	0.7941	0.6895	0.5052	0.5832
Decision tree	0.7588	0.5605	0.7139	0.6279
Naive Bayes	0.7588	0.6	0.6529	0.6253

strongly feel recall and F1 score are most appropriate as they give strong preference to the true positives, which are crucial for this dataset.

We find that the Decision Tree performs significantly better due to its recall value of 0.7139 and a higher F1 score of 0.6279. Naive Bayes almost has a similar performance as the Decision Tree but has a more balanced precision and recall score. This is particularly significant for our case, as we are mainly interested to understand what makes a recipient open a mail and the factors responsible for it, thus adding more weightage in correctly identifying the positive examples. All the codes used in this study are available at Github.[2]

6 Conclusion

The analysis of email-related behavior as a function of message and user characteristics is important for understanding the computer-mediated communication technology as well as for the development of automated tools to help people manage their email. We perform a detailed email interactivity study and observe certain similar patterns as mentioned in the existing studies. The recency of the dataset adds to the effectiveness and usefulness of this study. Then, we propose a novel methodology of feature selection where we perform dataset profiling using k-means clustering, based on the characteristics of the recipient as well as the day the mail is sent. We train three distinct binary classification models using Logistic Regression, Naive Bayes, and Decision Tree to predict whether an email recipient would open the email or not. We also ranked the features of each model in terms of its variable importance, calculated from its corresponding information gain. We compare the performance of the classifiers over accuracy, precision, recall, and F1 score. Decision Tree performs the best with a F1 score of 0.6279 and observe that the most significant feature to this model is the fraction of emails opened by the user in the past. We also observe that the training models achieve high performance, without applying any kind of boosting or ensemble methods.

[2] https://github.com/roysoumya/email-interactivity.

References

1. Q. Ai, S.T. Dumais, N. Craswell, D. Liebling, Characterizing email search using large-scale behavioral logs and surveys, in *Proceedings of the 26th International Conference on World Wide Web. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland* (2017), pp. 1511–1520. https://doi.org/10.1145/3038912.3052615
2. L.A. Dabbish, R.E. Kraut, Email overload at work: an analysis of factors associated with email strain, in *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, CSCW '06* (ACM, New York, NY, USA, 2006), pp. 431–440. https://doi.org/10.1145/1180875.1180941
3. L.A. Dabbish, R.E. Kraut, S. Fussell, S. Kiesler, Understanding email use: predicting action on a message, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '05* (ACM, New York, NY, USA, 2005), pp. 691–700. https://doi.org/10.1145/1054972.1055068
4. D. Di Castro, Z. Karnin, L. Lewin-Eytan, Y. Maarek, You've got mail, and here is what you could do with it!: analyzing and predicting actions on email messages, in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16* (ACM, New York, NY, USA, 2016), pp. 307–316. https://doi.org/10.1145/2835776.2835811
5. T.W. Jackson, A. Burgess, J. Edwards, A simple approach to improving email communication. Commun. ACM **49**(6), 107–109 (2006). https://doi.org/10.1145/1132469.1132493. Jun
6. T.W. Jackson, R. Dawson, D. Wilson, Understanding email interaction increases organizational productivity. Commun. ACM **46**(8), 80–84 (2003). https://doi.org/10.1145/859670.859673. Aug
7. F. Kooti, L.M. Aiello, M. Grbovic, K. Lerman, A. Mantrach, Evolution of conversations in the age of email overload, in *Proceedings of the 24th International Conference on World Wide Web. WWW '15, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland* (2015), pp. 603–613. https://doi.org/10.1145/2736277.2741130
8. K. Kushlev, E.W. Dunn, Checking email less frequently reduces stress. Comput. Human Beh. **43**, 220–228 (2015). https://doi.org/10.1016/j.chb.2014.11.005
9. W.E. Mackay, Diversity in the use of electronic mail: a preliminary inquiry. ACM Trans. Inf. Syst. **6**(4), 380–397 (1988). https://doi.org/10.1145/58566.58567. Oct
10. J.R. Tyler, J.C. Tang, When can i expect an email response? a study of rhythms in email usage, in *ECSCW 2003*, ed. by K. Kuutti, E.H. Karsten, G. Fitzpatrick, P. Dourish, K. Schmidt (Springer, Netherlands, Dordrecht, 2003), pp. 239–258
11. S. Whittaker, V. Bellotti, J. Gwizdka, Email in personal information management. Commun. ACM **49**(1), 68–73 (2006). https://doi.org/10.1145/1107458.1107494. Jan
12. L. Yang, S.T. Dumais, P.N. Bennett, A.H. Awadallah, Characterizing and predicting enterprise email reply behavior, in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17* (ACM, New York, NY, USA, 2017), pp. 235–244. https://doi.org/10.1145/3077136.3080782