Detection of Bot inorganic engagement on marketing emails using supervised machine learning algorithms

Moein Izadi, Suman Giri, Joe Brown, Chenyang Shi

Data Science & Analytics Solutions, Merck

Abstract

Customer email engagement is one of the most common and efficient ways for digital marketers and sales teams to reach health care professionals (HCPs). The data available from the email source systems (SFMC & Veeva) assumes all interactions are legitimate and represent the intention of the customer. However, there is a certain amount of "inorganic" email responses in the form of bots. Specifically, engagement actions such as, email opens and link clicks can be recorded, creating false positive responses.

Previously, two different approaches have been proposed at Merck to classify bot versus human engagement. First, a few time-related business rules based on domain knowledge of bot behaviors were proposed and implemented by a Data Science (DS) team in Germany. Second, another DS team from Netherland embedded invisible links to human eyes in SFMC emails called "Honeypot links" which can detect bot clicks accurately. These two approaches can classify email Click and Open activities as human or bot that could be used as training dataset. In other words, the two approaches can enable data labeling and leveraging supervised learning methods to readily classify emails as *open* and *click* activities into bot and human.

Building on the previous work from Merck Germany and Netherland Data Science teams, in the proposed method, different state-of-the-art machine learning algorithms are adopted to detect bot activities with high accuracy and reliability. The proposed module will learn from data generated from the source system and classify interaction as human or bot. This will enable better accuracy in downstream system to improve customer journey capabilities, channel affinity predictions, and next best action capabilities.

Keywords: Digital marketing, bot detection, Machine Learning classification, Inorganic engagement

Introduction

Emails are an important mode of communication for marketing and engaging customers in a low-touch and efficient manner. With exponential progress on data collection and data technologies such as data lakes/warehouses and cloud computing platforms, it is crucial for the companies to make data-driven decisions and focus on enhancing the customer journey. Unlike rule-based models, Machine learning models can facilitate decision making and fill the gaps when direct consumer data is not available. ML models generate insights to make accurate predictions about consumer behavior by analyzing historical data, identifying correlations, patterns and trends between data points. Rule based model's foundation is based on assumptions on bot behaviors that comes from limited domain and expert knowledge. On the other hand, ML supervised learning models can learn directly from limited historical data and be used for prediction on vast amount of new data. Email related data has been a great source for solving business and marketing problems using machine learning (ML) methods and techniques. For email service providers, knowledge of consumer interactions allows building predictive ML models for actions such as weather an email will be replied to or not [1]. The authors of [2] proposed that historical email data allows the prediction of user actions upon email delivery (e.g., marking it for deletion) [3]. Email marketing is predicted to have 10% annual growth and most enterprise marketers take advantage of it as a delivery channel tool [4,5]. However, compared to personal emails at

work or among friends, the organic (user) engagement is low. Organic in marketing is "pull" - initiatives that attract your targets (e.g., your brand). Inorganic marketing is "push" - initiatives that nudge your target to you (e.g., a digital ad). Moreover, bots or inorganic open and click activities are recorded and even contribute to much less than organic activities. Inorganic activities comprise almost 4% of total activities based on Honeypot observations. However, it plays an important role in digital marketing and the quality of interaction with customers. The open rates for the marketing email ranging from 15-30% depending on the industry type [6]. Thus, digital marketers have been trying to develop techniques that able them to enhance the engagement levels. In the proposed method, machine learning based models are proposed to detect bot versus human activities on open and click events.

Data and Methods

In this section, we mainly discuss and describe the Netherland data that was used for supervised learning methods. Before taking advantage of Netherland data for the proposed supervised learning method, France and Germany SFMC unlabeled data was investigated and a variety of supervised business rule-based, unsupervised, and semi-supervised labeling strategies on the unlabeled dataset was tested. In the best-case scenario, a supervised business rule-based model could mimic the parent business model performance. Thus, that's why having a dataset and a model independent from business rules was essential. The other unsupervised labeling strategies didn't provide us with a reliable dataset for modeling. The German business rule model and Netherland data are described as follow.

German data

It is a bot removal approach that proposes four business rules to label the email activities. First, they remove emails with common domains (e.g., "@yahoo.com", "@gmail.com") from bot- removal process. Useless click is recorded as the clicks on uncommon links to human in the sent emails. Then they remove instant bot activity which is an activity within less than a few minutes after email sent. Next, they remove delay bot activity (take the timestamp of useless click +/- X-minutes to get all clicks in that session). Lastly, they remove simultaneous activities which is removing click activities that happened at the same time (same timestamp + time differences since sent). Almost 60% of the initial data get removed throughout this proses.

Netherland data

The Netherland dataset was mainly used in this study. The data records are labeled based on the two aforementioned approaches; business rules and honeypot link deployments. There are three tables from Netherland, a dictionary, a table which contains the target feature (bot activity levels), and the last table contains other related features. The number of the entries in each table is 176876. And the disk size is less than 50 megabytes for the largest table.

The most important features (including independents), target feature levels, and a subset of data frame are shown and described in the tables below (1-3). Tables contain Marketing E-mail action fact data.

Table 1 Attributes Types and Descriptions

Attribute	Type	Description
Email	Text	E-mail address were the action or event (Sent/Open/Click…) belongs to.
Event Time	Date & Time	The date & time the action or event took place.
Event Type	Text	The type of the action or event. (Sent, Open, Click, Bounce…)
Browser	Text	The Brower from which the action (Open, Click) took place. This is not relevant for events (Sent, Bounce,…).
Device	Text	The device from which the action (Open, Click) took place. This is not relevant for events (Sent, Bounce,…).
Operating System	Text	The OS from which the action (Open, Click) took place. This is not relevant for events (Sent, Bounce,…).
Email Client	Text	The client from which the action (Open, Click) took place. This is not relevant for events (Sent, Bounce,…).
Is Unique	Boolean	Indication if the click was unique for the user, for the used newsletter.

Table 2 Data frame schema

Action Email	Event Time	Event Type	Browser	Device	Email Client	Is Unique	Operating System
louik.praet@azstjan.be	2019-11-21 13:06:53	Sent	-	-	-	-	-
heleena.coppenz@gza.be	2019-11-21 13:06:57	Click	Chrome	PC	Unspecified	True	Windows 7
heleena.coppenz@gza.be	2019-11-21 13:06:57	Open	Chrome	PC	Unspecified	True	Windows 7
heleena.coppenz@gza.be	2019-11-21 13:06:53	Sent	-	-	-	-	-

boris.deck@aszzz.be	2019-11-21 13:11:43	Click	Safari	iPhone	iPhone	True	iOS
boris.deck@aszzz.be	2019-11-21 13:11:27	Open	Safari	iPhone	iPhone	True	iOS

The table below shows the target feature levels.

Table 3 Target feature

Bot_Activity (Target)	Description
Bot_click_Honeypot	Click on honeypot link (honeypot goes over German. If a click is honeypot click, the German logic is not checked anymore for that link)
Bot_click_Honeypot_around	Clicks at the same time of the honeypot link click (As the clicks are at the same time, we assume they are bot clicks)
Bot_click_model	Bot click determined on the German bot model
Bot_open_model	Bot open determined on the German bot model
False	No bot clicks
N/A	No click

To implement the German rules, the Netherland team also did not use the common domains and took the standard links in the email footer as Useless link (unsubscribe, privacy, www.msd.nl). Application of the German rules and Honeypot links is simultaneously applied over all SFMC data, whereas bot activity in the Veeva data (AE) is solely determined by the German rules and was excluded for the further processing. So far, only SFMC mailings have integrated the honeypot link. All SFMC mailings from 31-03-2021 onward have Honeypot (HP) link integrated.

Data preparation

In this section, the data for ML modeling is discussed. First, we look at individual users' experience by sorting the data with email and event date (sent/open/click timestamps) as keywords. As shown below, the first entries of a certain user are displayed.

ME Action Email	ME Event Time	ME Event Type	ME Event Date	BOT_ACTIVITY	SentId	email_address
"k.d'hauwers@uro.umcn.nl"	2020-06-23 09:26:51	Sent	2020-06-23 09:26:51	false	NL|1489856|132018|2020-06-23	"k.d'hauwers@uro.umcn.nl"
"k.d'hauwers@uro.umcn.nl"	2020-11-04 07:00:48	Sent	2020-11-04 07:00:49	false	NL|1766941|132018|2020-11-04	"k.d'hauwers@uro.umcn.nl"
"k.d'hauwers@uro.umcn.nl"	2020-11-04 10:26:24	Open	2020-11-04 10:26:24	false	NL|1766941|132018|2020-11-04	"k.d'hauwers@uro.umcn.nl"
"k.d'hauwers@uro.umcn.nl"	2020-11-04 10:26:24	Click	2020-11-04 10:26:24	false	NL|1766941|132018|2020-11-04	"k.d'hauwers@uro.umcn.nl"
"k.d'hauwers@uro.umcn.nl"	2021-07-20 10:04:05	Sent	2021-07-20 10:04:06	false	NL|2354993|132018|2021-07-20	"k.d'hauwers@uro.umcn.nl"

Figure 1

In this figure 1, there are three unique "SentIds", suggesting three emails were sent to the user at three different times. The first row and last row only have the event type Sent, implying no response/reaction from users or bots. As a result, these two rows provide no useful information for modeling. In contrast, the middle three entries have Sent, Open and Click associated with one Sent Id, and two useful events can be derived as shown in the table below. Now we have separate columns for event types and timestamps for Sent and Click/Open, respectively, which can be useful to derive time-related features for machine learning.

ME Action Email	ME Event Time_x	ME Event Type_x	BOT_ACTIVITY_x	SentId	ME Event Time_y	ME Event Type_y
"k.d'hauwers@uro.umcn.nl"	2020-11-04 07:00:48	Sent	false	NL|1766941|132018|2020-11-04	2020-11-04 10:26:24	Click
"k.d'hauwers@uro.umcn.nl"	2020-11-04 07:00:48	Sent	false	NL|1766941|132018|2020-11-04	2020-11-04 10:26:24	Open

Figure 2

Next, we move on to the joining of tables. The steps are outlined as below:

1. Separate data into three tables named Sent, Open and Click.
2. Left join Sent and Click on email address and sent id.
3. Left join Sent and Open on email address and sent id.
4. Concatenate (2) and (3) vertically and remove any NAs in "Event Type_y" columns (this corresponds to the case where there is only SENT while no CLICK nor OPEN).

Considering activities after March 31, 2021, when the honeypot links were implemented, we have a data frame of 23534 entries (with 3716 unique emails, out of total 13140). The breakdown of the labels is shown below.

Table 4 Response feature levels

False	21037
Bot_click_model	929
Bot_open_model	649
Bot_click_Honeypot	470
Bot_click_Honeypot_around	449

The true labels from honeypot (470 + 449) accounts for ~3.9% of total entries.

Feature engineering

Since bot behavior and activities correlate strongly with time, extracting time-related features is prioritized. Three additional columns, "Sent_to_Open", "Sent_to_Click", and "Open_to_Click", for every event activity is calculated, which records the time differentials (in seconds) between each paired event types. According to domain knowledge and Exploratory Data Analysis (EDA) on data, these are the most important features that could potentially capture bot behaviors. To do so, we need to rearrange the table. In other words, we need to have three more columns added to the previous data frame which correspond to Sent/Open/Click time for every single record/activity.

One typical example is as below. Let's pick up one user (one email). This user has 8 unique sent ids, i.e. received emails at 8 separate time stamps. His/her or bot's reaction times of OPEN and CLICK are recorded in the last two columns.

If the reaction is OPEN, then we record the OPEN time with a CLICK time of NaT. If it is a CLICK, record the CLICK time and inherit the previous most recent OPEN time for OPEN time. Based on these three new columns, desired time differentials can be generated.

SentId	ME Action Email	Type_FROM_CLICK_OR_OPEN	ME Event Date_FROM_SENT	ME Event Date_FROM_CLICK_OR_OPEN	Sent_Time	Open_Time	Click_Time
NL|2108171|103847|2021-04-05	a.becker@vumc.nl	Open	2021-04-05 13:00:38	2021-04-05 13:12:29	2021-04-05 13:00:38	2021-04-05 13:12:29	NaT
NL|2211359|103847|2021-05-16	a.becker@vumc.nl	Open	2021-05-16 10:01:38	2021-05-16 10:58:11	2021-05-16 10:01:38	2021-05-16 10:58:11	NaT
NL|2275212|103847|2021-06-13	a.becker@vumc.nl	Open	2021-06-13 10:00:50	2021-06-13 10:03:55	2021-06-13 10:00:50	2021-06-13 10:03:55	NaT
NL|2478158|103847|2021-09-12	a.becker@vumc.nl	Open	2021-09-12 10:00:51	2021-09-12 10:39:33	2021-09-12 10:00:51	2021-09-12 10:39:33	NaT
NL|2546116|103847|2021-10-10	a.becker@vumc.nl	Open	2021-10-10 08:00:45	2021-10-10 08:26:02	2021-10-10 08:00:45	2021-10-10 08:26:02	NaT
NL|2555383|103847|2021-10-14	a.becker@vumc.nl	Open	2021-10-14 10:00:51	2021-10-14 14:03:47	2021-10-14 10:00:51	2021-10-14 14:03:47	NaT
NL|2585828|103847|2021-10-28	a.becker@vumc.nl	Open	2021-10-28 10:03:44	2021-10-28 11:15:34	2021-10-28 10:03:44	2021-10-28 11:15:34	NaT
NL|2585828|103847|2021-10-28	a.becker@vumc.nl	Click	2021-10-28 10:03:44	2021-10-28 11:15:46	2021-10-28 10:03:44	2021-10-28 11:15:34	2021-10-28 11:15:46
NL|2585828|103847|2021-10-28	a.becker@vumc.nl	Click	2021-10-28 10:03:44	2021-10-28 11:15:58	2021-10-28 10:03:44	2021-10-28 11:15:34	2021-10-28 11:15:58
NL|2585828|103847|2021-10-28	a.becker@vumc.nl	Click	2021-10-28 10:03:44	2021-10-28 11:15:58	2021-10-28 10:03:44	2021-10-28 11:15:34	2021-10-28 11:15:58
NL|2610561|103847|2021-11-07	a.becker@vumc.nl	Open	2021-11-07 10:00:57	2021-11-07 21:31:39	2021-11-07 10:00:57	2021-11-07 21:31:39	NaT

Figure 3

Engineered features on Sent_Time, Click_Time, Open_Time are described in the table below;

Table 5 Engineered features

Engineered Features	Type	Description
'Sent_to_Open', 'Sent_to_Click', 'Open_to_Click'	Numeric	Desired time differentials features
'Reference_time_to_Open', 'Reference_time_to_Click',	Numeric	A reference time (01/01/2021) was used for three more differential features to capture potential short to long term behaviors

'Reference_time_to_Sent'		
'Is_immidiate_Open', *'Is_immidiate_Click'*	Boolean	Whether an activity is within a few second (5 sec)

Results

For the modeling part, Pycaret package was deployed which is an open-source, low-code machine learning library in Python. It automates machine learning workflows. The model and all the steps from EDA and preprocessing to the modeling was performed on AWS SageMaker cloud space. First, due to highly imbalanced data (honeypot labels account for only ~4% of the total data), two different anomaly detection algorithms (one-class SVM and Isolation forest) were applied to the data. However, bots as outliers/anomalies were not detected efficiently. The second approach was a supervised multi-class classification problem. In the table below, 13 models based on different metrics (best results are highlighted) using 6 numeric (feature engineered described in table 4) and 6 categorical features mentioned in table 2 can be compared. For example, Random Forest with 100 trees achieved the highest performance. For training-test split, 30-70% proportion was considered. The results displayed after stratified 10-fold Cross Validation, removing multicollinearity and perfect collinearity.

Table 6 Comprehensive models performance

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Random Forest	0.9658	0.986	0.8506	0.9664	0.9659	0.8291
Light Gradient Boosting	0.9654	0.9929	0.8428	0.9655	0.9652	0.8242
Decision Tree	0.9634	0.9282	0.8479	0.9647	0.9638	0.8191
Gradient Boosting	0.9596	0.9888	0.8092	0.9593	0.9587	0.7917
Extra Trees Classifier	0.9588	0.9676	0.8224	0.9592	0.9589	0.7933
K Neighbors	0.9486	0.9758	0.7953	0.9505	0.9492	0.7465
SVM - Linear Kernel	0.8938	0	0.2	0.7988	0.8436	0
Ridge Classifier	0.8933	0	0.2437	0.8371	0.8565	0.1348
Logistic Regression	0.8921	0.8056	0.2075	0.8048	0.845	0.0294
LDA	0.8712	0.9085	0.4921	0.8851	0.8767	0.4121
Ada Boost	0.8612	0.8362	0.5145	0.8973	0.8733	0.4009
QDA	0.7129	0.7603	0.3852	0.8762	0.7456	0.2727
Naive Bayes	0.6798	0.7806	0.4875	0.8536	0.7429	0.1711

After removing German labels and modeling solely with the Honeypot labels to have an independent model from business rules, the number of records dropped to ~22k (binary classification). Here are the results of top three models for 22k dataset.

Table 7 Three top models for 22k dataset

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
Light Gradient Boosting	0.996	0.998	0.94	0.95	0.95	0.946
Decision Tree Classifier	0.995	0.97	0.94	0.93	0.94	0.93
Random Forest Classifier	0.99	0.999	0.93	0.94	0.94	0.937

ROC curve is demonstrated in the figure below for Random Forest model. Class imbalance is causing such an amazing AUC due to the fact that the model is predicting the overrepresented class the majority of the time.

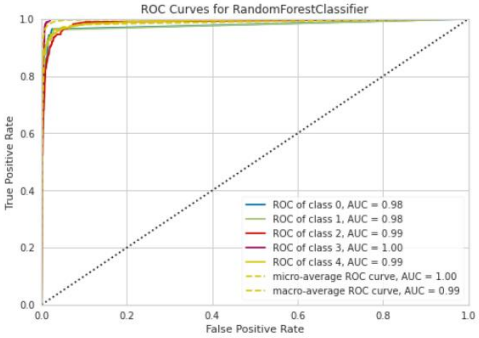

Figure 4 ROC Curve for Random Forest Classifier.

Discussion

As we can see, tree-based models show the best performance. However, a very high model performance could potentially come from data or model *leakage*! In addition, another reason for unrealistic high test performance could be that test dataset does not contain the same proportion of positive-labeled data points as training set. Therefore, we need to make sure the models are trained and tested appropriately. To avoid the latter issue, shuffling and stratified k-fold Cross Validation were considered in the training phase. "*Stratified cross-validation is a great technique in the case of highly imbalanced classes. For binary classification with a training/test split rather than cross-validation, this involves the training set having the same proportion (50-50) of positive-labeled points as the test set (and hence the same as the overall training set)[7]*. In the next step, we investigate the potential model leakage.

Leakage investigation

There could be a few sources of leakage that a model learns from the patterns/info in the test dataset as well. For our case, the source of leakage could be the multiple clicks on the same link for the same email (Bot_honeypot_around), or leakage of a portion of activities from the same email and same "SentId" into the test set. As it was mentioned, to make our analysis simpler, first the dataset was split into Click – Open (17353 and 6181 records). Open target feature contains *German* labels and Click target contains *German*, *"Bot_click_Honeypot"*, and *"Bot_click_Honeypot_around"*. Comprehensive analysis of different ML models on the split data is shown below. OPEN is shown on the top and CLICK on the bottom.

Table 8 Top two models for Open and Click tables separately.

Model (Open)	Accuracy	AUC	Recall	Prec.	F1	Kappa
Gradient Boosting	0.989	0.997	0.992	0.997	0.994	0.862
Decision Tree	0.989	0.967	0.993	0.996	0.994	0.859
Model (Click)	**Accuracy**	**AUC**	**Recall**	**Prec.**	**F1**	**Kappa**
Random Forest	0.89	0.96	0.81	0.89	0.89	0.77
Decision Tree	0.88	0.90	0.82	0.89	0.88	0.77

It's also a common practice to consider the baseline model performance as well. The baseline accuracy for OPEN model is (16704) / (16704 + 649) = **96.26%**; And for CLICK model is (4333) / (4333 + 929 + 470 + 449) = **70.10%**

One of the best practices to detect leakage is to have a validation (unseen) dataset for a sanity check on Open and Click tables, separately. The assumption is that this data hasn't gone through the same preprocessing. First, a small portion of the sorted Open table was selected as the held-out validation, considering the ~4% proportion of the minority class. The validation and the train-test sets don't have any common emails.

Hereunder the feature importance plot and confusion matrix for the validation prediction (unseen) are depicted if figure 4 and Table 9. For Click table, same strategy was adopted where German and *"Bot_honeypot_around"* were removed to make the analysis more reliable. Here is the feature importance plot.

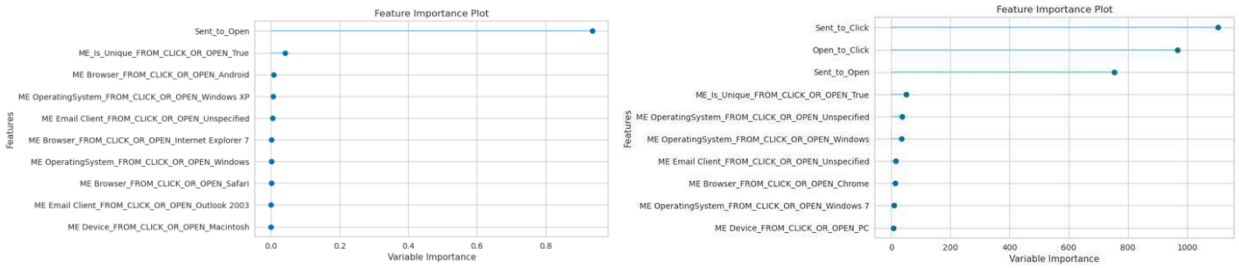

Figure 5 Feature importance plot for Open and Click tables.

The prediction on the Click validation set with the same characteristics as the Open validation is demonstrated in the Table 9 below. The confusion matrix for the validation set shows a high performance on unseen email activities (Table 9). "Sent to Open" in Open table and "Sent to Click", "Sent to Open" and "Open to Click" for the Click table show the highest feature importance in bot detection (figure 4).

Table 9 Confusion matrix for validation

Actual \ Predicted	Bot_open_model	False
Bot_open_model	220	7
False	58	5068

Actual \ Predicted	Bot_click_Honeypot	False
Bot_click_Honeypot	48	11
False	14	730

Conclusion

In this study, bot versus human *open* and *click* activities on SFMC emails were investigated and modeled using different business rule based and state-of-the-art machine learning methods. Almost 4% of email activities are inorganic activities according to the clicks on Honeypot links and the proposed ML models could predict most of them efficiently. Inspired by business rules and EDA, time extracted feature engineering such as time differential among Sent, Open and Click times was conducted and improved the model's performance drastically. This is projected in the feature important plots on Open and Click tables separately. Ensemble tree-based models demonstrated the best performance compared to other methods on the most of metrics. Due to high performance of ML models, data and model leakage was investigated and unseen data with unique emails was used for model validation. Even the model's performance on the unseen validation sets was stunning. For this case study, we are more interested to see and detect inorganic activities. If an activity is labeled as human but it's a bot, then we will not be able to send reminding emails to keep the level of engagement. The proposed models with high performance on precision, recall and F-score can be implemented for production to detect bot activities on emails efficiently and increase the quality of interactions with companies' clients. In the future works, a comprehensive comparison between business rule-based models and proposed ML model can be conducted.

Acknowledgment

Hereby, we sincerely thank "Roel van Maris" and "Duco de Beus" from Netherland data science team who provided us with the labeled dataset and dictionary. And also, we would like to thank Jeevaka Kiriella and Yishu Gong for their constructive feedback/comments on the manuscript that helped to improve the paper.

Reference

Liu Yang, Susan Dumais, Paul Bennett, and Ahmed Hassan Awadallah. 2017. Characterizing and Predicting Enterprise Email Reply Behavior. In Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). ACM.

Dotan Di Castro, Zohar Karnin, Liane Lewin-Eytan, and Yoelle Maarek. 2016. 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, 307–316. https://doi.org/10.1145/2835776.2835811

Laura Dabbish, Gina Venolia, and JJ Cadiz. 2003. Marked for Deletion: An Analysis of Email Data. In CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03). ACM, New York, NY, USA, 924–925. https://doi.org/10.1145/765891.766073

G Tsirulnik. 2011. British Airways mobile email campaign garners 250K app downloads. http://www.mobilemarketer.com/ex/mobilemarketer/cms/news/email/9056.html. (2011).

Shar VanBoskirk, CS Overby, and S Takvorian. 2011. US interactive marketing forecast 2011 to 2016, Forrester Research. (2011).

D Wells. 2016. Email marketing statistics 2017. http://www.smartinsights.com/email-marketing/email-communications-strategy/statistics-sources-for-email-marketing/. (2016).