# Behavior Toward Newcomers and Contributions to Online Communities

Florian Pethig* Hartmut Hoehle† Kai-Lung Hui‡ Andreas Lanz§

**Online Appendix**



*Tilburg School of Economics and Management, Tilburg University. Email: f.pethig@tilburguniversity.edu

†Business School, University of Mannheim. Email: hoehle@uni-mannheim.de

‡School of Business and Management, Hong Kong University of Science and Technology. Email: klhui@ust.hk

§Faculty of Business and Economics, University of Basel. Email: andreas.lanz@unibas.ch

# A Data Collection and Preparation

We constructed our data in two rounds. In Round 1, between August and October 2018, we retrieved all deals that were available on mydealz, and our analysis of these deals revealed that the platform broadly adopted the newcomer nudge on October 20, 2016. Initially, we faced the challenge of identifying *exactly* the first deal posted by a new member before the policy change because, apart from the nudge, the first deals do not differ from any other deals. Fortunately, after our first round of data collection, mydealz updated old newcomer deals to show the nudge. In Round 2, in September 2019, we collected the deals again and combined the results of both rounds.

Table A1 outlines our 180-day sample of the two rounds of data collection, divided into six aggregated 30-day intervals. As shown in Table A1, the platform started to gradually test the nudge before October 20. From July 22 to October 19, the nudge was available on fewer than 1% of deals (0.03% to 0.87%). After the intervention was widely introduced, the nudge was available on 11.29% to 12.85% of the deals. As expected, in Round 2, the proportion of deals that carried the nudge was comparable before and after the intervention. Interestingly, some deals had the nudge in Round 1 but not in Round 2, leading to the assumption that mydealz initially identified some deals as newcomer deals that, in fact, were not truly newcomer deals. Finally, the last column of Table A1 shows that 0.16% to 0.32% of the deals were removed by the platform after Round 1 because they were flagged as spam or duplicates (Figure A1).

To identify suitable newcomer deals (treatment group) and non-newcomer deals (control group), we modified our sample in several ways. First, we removed 119 deals that were removed from the platform after Round 1. Second, we restricted our sample to deals by posters who had not deleted their profile at the time of data collection, had not been banned for violating the rules, and were not employed by the community (e.g., as a deal-hunter, moderator, or administrator). Third, the platform had tested the nudge on select deals before its widespread introduction. We removed these deals from our main analysis to establish clean pre- and posttreatment periods across our treatment and control groups. Fourth, we only kept the newcomer deals for which the nudge was not removed before the second round of data collection to ensure that our treatment group consisted of legitimate newcomer deals. Table A2 shows our final sample, comprising 4,952 newcomer deals in the treatment group and 35,971 non-newcomer deals in the control group. The sum of the bold numbers in Columns (6) and (7) shows the number of newcomer deals, and the sum of Column (8) shows the number of non-newcomer deals.

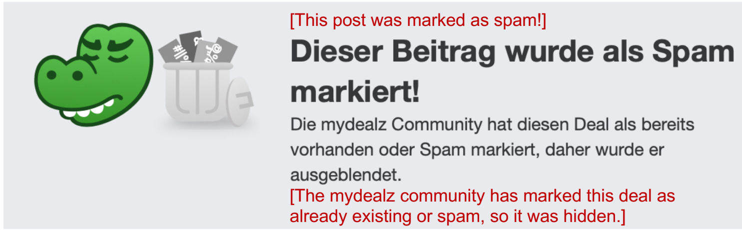


Figure A1: Screenshot of a Deal Marked as Spam

Table A1: Data Collection Strategy

| | Round 1 *(R1)* | | | Round 2 *(R2)* | | | |
|---|---|---|---|---|---|---|---|
| Time periods | Deals | Deals with nudge | % of deals with nudge | Deals | Deals with nudge | % of deals with nudge | % of deals removed |
| day30 | | | | | | | |
| Jul 22 – Aug 20 | 7,420 | 2 | 0.03 | 7,406 | 621 | 8.39 | 0.19 |
| Aug 21 – Sep 19 | 7,464 | 48 | 0.64 | 7,452 | 730 | 9.80 | 0.16 |
| Sep 20 – Oct 19 | 7,930 | 69 | 0.87 | 7,913 | 799 | 10.10 | 0.21 |
| Oct 20 – Nov 18 | 7,774 | 878 | 11.29 | 7,761 | 783 | 10.09 | 0.17 |
| Nov 19 – Dec 18 | 13,055 | 1,677 | 12.85 | 13,013 | 1,542 | 11.85 | 0.32 |
| Dec 19 – Jan 17 | 9,278 | 1,065 | 11.48 | 9,257 | 978 | 10.56 | 0.23 |

*Note:* Black Friday was on Nov 25, 2016. Many stores offer highly promoted sales on (and after) this day. Thus, 68% more deals were posted between Nov 19 and Dec 18 compared to the previous 30 days.

Table A2: Sample Selection Procedure

| | Deal removed after R1? | | Deal posted by deleted, banned, or employed member? | | Deal has nudge? | | | |
|---|---|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) |
| Time periods | Yes | No | Yes | No | R1 \ R2 | R2 \ R1 | R1 ∩ R2 | No |
| Jul 22 – Aug 20 | 14 | 7,406 | 1,726 | 5,680 | 0 | **584** | 1 | **5,095** |
| Aug 21 – Sep 19 | 12 | 7,452 | 1,623 | 5,829 | 21 | **650** | 23 | **5,135** |
| Sep 20 – Oct 19 | 17 | 7,913 | 1,875 | 6,038 | 22 | **703** | 39 | **5,274** |
| Oct 20 – Nov 18 | 13 | 7,761 | 1,833 | 5,928 | 86 | 4 | **699** | **5,139** |
| Nov 19 – Dec 18 | 42 | 13,013 | 2,434 | 10,579 | 123 | 0 | **1,409** | **9,047** |
| Dec 19 – Jan 17 | 21 | 9,257 | 1,992 | 7,265 | 76 | 1 | **907** | **6,281** |

*Note:* Numbers highlighted in bold represent the deals included in the main analysis. R1 \ R2 = Nudge present in R1 but not in R2. R2 \ R1 = Nudge present in R2 but not in R1. R1 ∩ R2 = Nudge present in R1 and R2.

# B Construction of Additional Variables

## B.1 Price Comparison and Discount

We use the average likelihood with which users mention a price comparison in the descriptions of their subsequent deals, *AvgPriceComp*, and the average percentage discount, *AvgDiscount*, as alternative measures of the quality of subsequent deals. According to the official mydealz community guidelines,[1] deals should include a price comparison so that users can objectively assess their savings potential, allowing us to construct the aforementioned measures of deal quality: (1) if a price comparison is mentioned in the deal description, this indicates that the deal poster was aware of the guidelines and tried to adhere to them by addressing this aspect in his or her post; (2) if the comparison also includes the comparison price, this allows us to calculate the savings potential of a deal.[2] Unfortunately, for the deals in our study period, such data are not available in any kind of structured format. At that time, deals typically only had the discounted price at the top and the comparison price, if available, was mentioned in the text (Figure B1). Only after our study period, mydealz made it mandatory to enter the comparison price along with the discounted price.

To address this limitation, we use a regular expression (regex) to extract the keywords price comparison ("Preisvergleich"), comparison price ("Vergleichspreis"), their abbreviations ("PVG", "VGP"), and the names of two platforms for conducting a price comparison in Germany ("Idealo" , "Geizhals") from the description of all deals. The regex patterns that we use are shown in Table B1. Of the 11,757 newcomers and non-newcomers who posted at least one additional deal within 12 months, 46% mentioned a price comparison in the deal description of the subsequent deals.

To extract the comparison price (and not just mentions of it), we relied on the same regex described above and extracted any euro amount (€|Euro|euro|EUR|eur) that was mentioned in the 100 characters trailing the occurrence of the word "price comparison" (or any variants thereof). The intuition for this approach is demonstrated in Figure B1—the word "price comparison" is typically followed by the actual amount (both circled in red). We divided the deal price (available in a structured form from the field at the top) by the comparison price to obtain the discount percentage. The resulting average discount percentage for subsequent deals was 29.6%.[3]

[1] See https://www.mydealz.de/faq-nutzung#deals-einstellen (in German).

[2] Mentioning the price comparison does not automatically entail that the price is also mentioned. For example, a deal poster may mention a price comparison if it is not available, e.g., because the deal is a product available only in a certain shop.

[3] We set values to missing where the extracted comparison price was higher than the deal price.

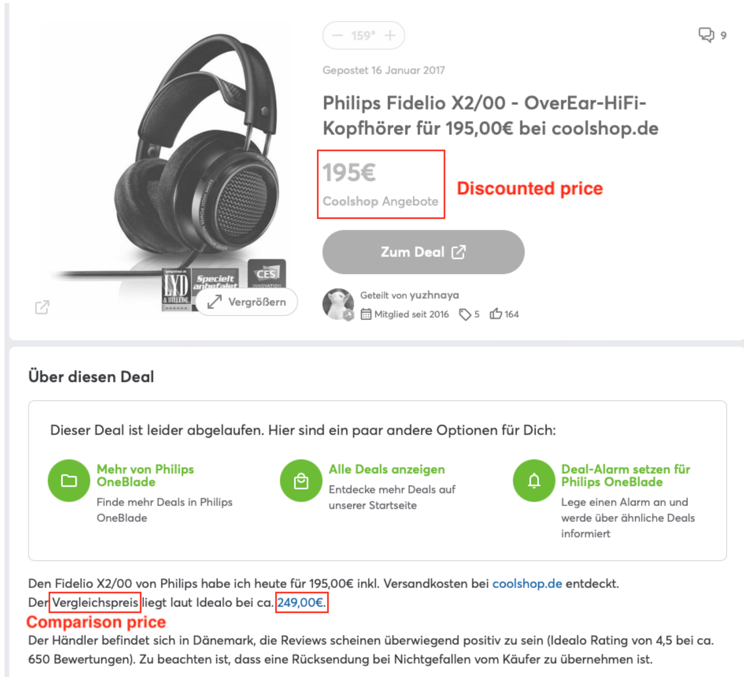


Figure B1: Screenshot of a Deal from January 2017

Table B1: List of Regex Patterns to Filter Price Comparisons

| English Keyword | German Keyword (Abbreviation) | Regular Expression |
|---|---|---|
| Price comparison | Preisvergleich (PVG) | `(p|P)reisvergleich|(p|P)(v|V)(g|G)` |
| Comparison price | Vergleichspreis (VGP) | `(v|V)ergleichspreis|(v|V)(g|G)(p|P)` |
| Next best price | Nächster Preis | `(n|N)ächste(r?)(p|P)reis` |
| Idealo | Idealo | `(i|I)dealo|IDEALO` |
| Geizhals | Geizhals | `(g|G)eizhals|GEIZHALS` |

## B.2 Percentage of Negative Words

We use the percentage of negative words, *PercNegWords*, as an alternative measure of the sentiment of comments posted for a given deal (e.g., Shen et al., 2015). We tokenized each comment, removed punctuation, and lowercased and matched the words with the negative word list of SentiWS (Remus et al., 2010), a popular German sentiment lexicon, which has been shown to perform particularly well for negative words (Sidarenka & Stede, 2016).[4] To construct the measure of *PercNegWords* at the deal level, we divide the sum of negative words across all comments for deal $i$ by the sum of all words across all comments for deal $i$.

[4] We added the English word "cold" to the SentiWS negative word list because the word is frequently used to label bad ("cold") deals.

## B.3 Dictionary-Based Sentiment Score

As another robustness check, we conducted a dictionary-based sentiment analysis on an English version of the comments. We used the "deep_translator" package in Python, which provided us with access to the Google Translate API, to translate all comments within 30 days before and after the policy change from German to English.[5] We applied the "sentimentr" package in R on the translated comments and obtained one sentiment score for each sentence.[6] We averaged the scores at the deal level to obtain an aggregated score, *SentScore*, and reran Equation (1). The results shown in Table B2 are largely consistent, but smaller in magnitude.

Table B2: Results of the Dictionary-Based Sentiment Analysis

| | *SentScore* | | |
|---|---|---|---|
| | ±3 Days | ±5 Days | ±30 Days |
| *Newcomer* | 0.009 | −0.001 | 0.001 |
| | (0.014) | (0.012) | (0.006) |
| *Newcomer×After* | 0.013 | 0.031* | 0.012* |
| | (0.021) | (0.018) | (0.007) |
| *Tenure* | 0.000 | 0.000 | 0.000** |
| | (0.000) | (0.000) | (0.000) |
| log(1+*DescLen*) | 0.002 | 0.006* | 0.007*** |
| | (0.005) | (0.004) | (0.001) |
| *LocalDeal* | 0.013 | 0.016* | 0.005* |
| | (0.011) | (0.009) | (0.003) |
| *NumCategories* | 0.000 | 0.000 | 0.000 |
| | (0.002) | (0.001) | (0.001) |
| *Content* | 0.000 | 0.015 | 0.006 |
| | (0.022) | (0.016) | (0.005) |
| *AvgCommentLen* | 0.001 | 0.000 | 0.000 |
| | (0.000) | (0.000) | (0.000) |
| Time FE (Day, Hour) | Yes | Yes | Yes |
| Category FE | Yes | Yes | Yes |
| Observations | 1,065 | 1,748 | 11,234 |
| Adjusted $R$-squared | -0.004 | 0.009 | 0.006 |

*Note:* FE = fixed effects. Robust standard errors clustered by user are in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.10$.

[5] See https://pypi.org/project/deep-translator/.

[6] See https://cran.r-project.org/web/packages/sentimentr/sentimentr.pdf. We manually change the valence of the term "hot" from -0.25 to 0.5 because it reflects a positive sentiment in the context of the mydealz community.

# C Newcomers Revealing Themselves Versus the Nudge

We separated the identification of the newcomer ("This is the first deal by [user]") versus the suggested treatment of the newcomer ("Help out by posting tips or just thank them for their deal."). We constructed a new independent variable *FirstDealMentioned* that was set to 1 if a newcomer mentioned that it was his or her first deal in the deal description of deal $i$ and 0 otherwise.[7] In total, 651 newcomers (13%) disclosed their newcomer status. Thus, there were two treatments, *Newcomer*×*After* and *FirstDealMentioned*. Both variables show an impact if people changed their behavior because the poster was a newcomer. Only the interaction *Newcomer*×*After* (not *FirstDealMentioned*) would have an effect if people changed their behavior because of the platform's instruction instead of the newcomer's status.

The results in Column (1) of Table C1 show that the coefficient of *Newcomer*×*After* is positive and significant for number of comments. The coefficient of *FirstDealMentioned* is also positive and significant but smaller in magnitude. This finding indicates that self-disclosure may also attract more comments, but to a lesser extent than the nudge. Thus, the nudge may have been more effective in changing behavior, suggesting that people change their behavior because of the platform's instruction. The interaction *FirstDealMentioned*×*After* is not significant, suggesting that self-disclosure in combination with the nudge did not affect the number of comments.

In Column (2), we observe evidence consistent with the idea that the nudge is a more powerful intervention than self-disclosure. Whereas *Newcomer*×*After* had a positive and significant effect on *Positive*, *FirstDealMentioned* and *FirstDealMentioned*×*After* did not. In Columns (3) and (4), we find that *FirstDealMentioned* is negatively related to *Neutral* and positively related to *Negative*, which indicates increased polarization. This effect disappears after the introduction of the nudge.

Overall, this analysis helps to separate the identification of newcomer deals and the suggested treatment of asking established members to be nice to newcomers. *FirstDealMentioned* discloses the newcomer status, but not the message of the nudge. The nudge combines the two. So, including *FirstDealMentioned*×*After* should tease out these two effects. *Newcomer*×*After* should then provide a more precise estimation of the nudging effect.

[7] We used the following regular expression to extract *FirstDealMentioned* from newcomer deals: `(e|E)rste(r?)|1\.?)(d|D)eal` (essentially matching "first deal" or "1st deal" in German). As self-disclosure of the newcomer status was only relevant to newcomers, we applied the regular expression only to newcomer deals and set *FirstDealMentioned* to 0 for all non-newcomer deals.

Table C1: Newcomers Revealing Themselves Versus the Nudge

| | log(1+*NumComments*) | *Positive* | *Neutral* | *Negative* |
|---|---|---|---|---|
| | (1) | (2) | (3) | (4) |
| *Newcomer* | −0.381*** | −0.004 | 0.001 | 0.003 |
| | (0.030) | (0.005) | (0.005) | (0.005) |
| *Newcomer*×*After* | 0.441*** | 0.013** | −0.009 | −0.004 |
| | (0.035) | (0.006) | (0.006) | (0.005) |
| *FirstDealMentioned* | 0.140** | 0.006 | −0.025** | 0.019* |
| | (0.071) | (0.012) | (0.012) | (0.011) |
| *FirstDealMentioned*×*After* | −0.049 | −0.005 | 0.026* | −0.021 |
| | (0.087) | (0.015) | (0.015) | (0.013) |
| *Tenure* | 0.003*** | 0.000** | 0.000 | 0.000*** |
| | (0.001) | (0.000) | (0.000) | (0.000) |
| log(1+*DescLen*) | 0.158*** | 0.011*** | −0.010*** | −0.001 |
| | (0.012) | (0.001) | (0.002) | (0.001) |
| *LocalDeal* | −0.372*** | −0.018*** | 0.035*** | −0.017*** |
| | (0.024) | (0.003) | (0.003) | (0.002) |
| *NumCategories* | 0.037*** | −0.002*** | 0.002*** | 0.000 |
| | (0.004) | (0.000) | (0.000) | (0.000) |
| *Content* | −0.545*** | 0.002 | 0.000 | −0.001 |
| | (0.025) | (0.004) | (0.004) | (0.004) |
| *AvgCommentLen* | | 0.001*** | −0.003*** | 0.002*** |
| | | (0.000) | (0.000) | (0.000) |
| Time FE (Day, Hour) | Yes | Yes | Yes | Yes |
| Category FE | Yes | Yes | Yes | Yes |
| Observations | 40,923 | 39,307 | 39,307 | 39,307 |
| Adjusted $R$-squared | 0.139 | 0.031 | 0.102 | 0.064 |

*Note:* FE = fixed effects. Robust standard errors clustered by user are in parentheses. *** $p$ <0.01, ** $p$ <0.05, * $p$ <0.10.

# D Effect of Nudge on Alternative Retention Variables

We consider the effect of the nudge on alternative outcomes to measure the retention of newcomers. *NumDealsPosted* is a count measure of the number of deals posted by a user over the 12 months following the first deal. Column (1) of Table D1 shows that the nudge yields a 5% increase in the volume of deals by newcomers compared to non-newcomers. We also consider the effect of the nudge on commenting behavior. We use a binary indicator to determine whether a user posted any comments during the 12 months after the first deal, *CommentPosted*. The results using a linear probability model (LPM) are shown in Column (2) of Table D1. We show evidence that newcomers after the policy change were 7 percentage points more likely to post a comment compared to non-newcomers. We further differentiated whether the comment was posted to a deal posted by the commenter herself or by another community member. The former reflects a revisiting and refinement of their own content, whereas the latter reflects a shift to explore and discuss content generated by the community. These alternative dependent variables have appended suffixes: *Own* refers to a comment on a deal posted by the commenter and *Other* refers to a comment on a deal posted by another community member. In Columns (3) and (4) of Table D1, we present the results of regressions to estimate the decision to comment on a deal posted by the commenter herself, *CommentPostedOwn*, and a deal posted by another user, *CommentPostedOther*. The results indicate a positive effect on *CommentPostedOwn* and *CommentPostedOther*. However, the coefficient is larger for *CommentPostedOwn* than for *CommentPostedOther*. This indicates that newcomers were more likely to comment on their own content than to discuss others' content after the policy change.

Table D1: Effect of Nudge on Alternative Retention Variables

| | log(1+*NumDealsPosted*) | *CommentPosted* | *CommentPostedOwn* | *CommentPostedOther* |
|---|---|---|---|---|
| | (1) | (2) | (3) | (4) |
| *Newcomer* | −0.398*** | −0.145*** | −0.178*** | −0.167*** |
| | (0.021) | (0.011) | (0.013) | (0.012) |
| *Newcomer×After* | 0.046* | 0.074*** | 0.101*** | 0.050*** |
| | (0.025) | (0.013) | (0.015) | (0.014) |
| *Tenure* | −0.008*** | 0.001*** | 0.000** | 0.002*** |
| | (0.000) | (0.000) | (0.000) | (0.000) |
| *BadgeVote* | 0.092*** | 0.048*** | 0.041*** | 0.086*** |
| | (0.020) | (0.004) | (0.008) | (0.006) |
| *BadgeComment* | 0.167*** | 0.054*** | 0.124*** | 0.112*** |
| | (0.021) | (0.004) | (0.009) | (0.006) |
| *BadgeDeal* | 0.681*** | −0.007* | 0.070*** | −0.020*** |
| | (0.023) | (0.004) | (0.008) | (0.005) |
| Time FE (Day) | Yes | Yes | Yes | Yes |
| Observations | 19,153 | 19,153 | 19,153 | 19,153 |
| Adjusted $R$-squared | 0.190 | 0.095 | 0.089 | 0.147 |

*Note:* FE = fixed effects. Standard errors clustered by user are in parentheses. *** $p<0.01$, ** $p<0.05$, * $p<0.10$.

# E Qualitative Evidence

We interviewed 18 mydealz' members to corroborate our results.[8] The interviews were conducted in November and December 2021 and lasted 23 minutes on average. Participants were recruited through mydealz and deal-related Facebook groups. Six interviewees stated that they felt that the nudge would make them or other members more friendly toward newcomers[9] and that this would sometimes come at the cost of upvoting deals with minor flaws. For example, one interviewee stated:

> So if it says it's a newcomer, then the user has a bit of a 'puppy license' and I try to take the newcomer somehow in protection and maybe, although it is not such a great deal, still vote hot.

This notion of a "puppy license" (*Welpenschutz* in German) describes the special status of young puppies, i.e., a "leeway period granted by older members of the group" (Natterson-Horowitz & Bowers, 2020, p. 51). It exists for many animal species—and even humans—and helps new community members explore different behaviors without facing the same consequences as established members. One mydealz' editor also confirmed the efficacy of the nudge, when asked whether people would respond more positively:

> Yes, definitely yes. So if the deal is *really* [emphasis added] not good, then the user is also informed of that, but if there are minor errors, for example, a price comparison was forgotten, then it is usually just pointed out.

This quote again confirmed a "lenient" period for newcomers due to the nudge with the result that newcomers are not appraised using the same standards as established members. The qualitative evidence underscores that established members may have been more lenient than justified by the behavior of newcomers. Newcomers could not achieve the same score in the absence of the nudge.

[8] The interview guide is available at https://osf.io/t7awu.

[9] One additional interviewee stated that she had never noticed the nudge but that she was more lenient toward newcomers who self-disclosed their status.

## F Detecting Friendly, Helpful, Useful, and Informative Comments

We explore how the content of the comments changed in response to the nudge. We trained several supervised learning algorithms to detect friendly, helpful, useful, and informative comments. To obtain reliable training data, we instructed three research assistants who are native German speakers to label 4,000 comments on the dimensions of friendliness, helpfulness, usefulness, and informativeness.[10] The research assistants were at least somewhat familiar with mydealz. Two had registered user accounts, and one was actively participating on the platform by posting deals and comments. All comments were rated on 9-point semantic differential scales adapted from Wenninger et al. (2019) and Yin et al. (2014). To ensure sufficient variation in the friendliness of the labeled comments, we randomly selected 1,000 comments from deals with a deal temperature less than or equal to zero (25%) and 3,000 comments from deals with a deal temperature above zero (75%). Although only 5.4% of the comments in the data set were from deals with a low deal temperature, highly imbalanced data can be a challenge for machine learning algorithms (He & Garcia, 2009). According to a recent analysis on Stack Overflow (Punyon & Montrose, 2020), only 0.78% of the comments on the platform were labeled as unfriendly and these comments were more likely to be written in response to low-quality questions. Thus, human annotators received more comments from deals with a low net score. We did not expect this choice to influence the variation in helpful, useful, and informative comments because they might have been written in response to a high- or low-quality deal. We also provided human annotators the option to flag suspicious comments, e.g., when they looked truncated or were not understandable.

The annotation process was implemented using formr, an online tool that produces surveys based on comma-separated values (CSV) files (Arslan et al., 2020). We split the annotation task into 20 surveys containing 200 comments each (20 per page) and randomized the order of the comments (per page) to mitigate response-order effects. To match annotations across the surveys, the annotators entered a self-generated identification code at the beginning of each survey. We removed all comments that any of the annotators flagged or did not rate, leaving 3,915 fully annotated comments. We averaged the scores of the annotators and rounded the values to the nearest integer. For our classification task, we were primarily interested in detecting changes in the distribution of (1) friendly comments because they might encourage newcomers to stay and (2) helpful, useful, or informative comments because they might convey important information. Thus, we collapsed

[10] The annotation instruction (in English) provided to the research assistants is available at https://osf.io/t3fbh.

the 9-point semantic differential scales into binary scales. We coded friendly, helpful, useful, and informative comments (6, 7, 8, 9) as 1 and unfriendly, not helpful, not useful, and not informative comments (1, 2, 3) as 0. In addition to removing neutral comments with an average rating of 5, we also removed comments with an average rating of 4 because all three annotators rated the majority of the comments as not conveying important information (for a similar argument, see Liu et al., 2020). Including comments with an average rating of 4 led to highly imbalanced classes, which was detrimental to the classification performance. Thus, we did not include these comments in our classification task.

We pre-processed each comment by removing punctuation, digits, single characters, and stop words. The remaining words were lowercased, and the Snowball stemming algorithm was applied to reduce words to their stem.[11] We translated each resulting string into a term-frequency inverse document frequency (tf-idf) representation and applied six commonly used supervised learning algorithms, including gradient boosting, logistic regression, naïve Bayes, neural network, random forest, and support vector machine (Clarke et al., 2020). We adopted the scikit-learn package (Pedregosa et al., 2011) and divided the sample into training data (70%) and test data (30%) to evaluate the performance of each algorithm. Precision, recall, and $f$-measure were our evaluation criteria, and the results are shown in Table F1. The classification performance was in line with recent research that has classified posts in online knowledge communities (Liu et al., 2020).

We selected the machine learning algorithm with the highest $f$-measure (support vector machine for friendly comments, gradient boosting for helpful, useful, and informative comments) to classify all of the remaining comments. According to the resulting machine learning classifications, the percentage of friendly (*PercFriendComments*), helpful (*PercHelpComments*), useful (*PercUseComments*), and informative comments (*PercInfoComments*) were constructed as dependent variables. The results in Table F2 suggest that the nudge did not result in any significant change in the quality of the information provided by the comments.[12]

[11] The Python code for text pre-processing is available at https://osf.io/e3huy.

[12] The resulting sample size is lower because after pre-processing some comments were empty strings and could not be classified.

Table F1: Performance of Machine Learning Algorithms

| | Friendly | | Helpful | | Useful | | Informative | |
|---|---|---|---|---|---|---|---|---|
| Measure | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| **Gradient Boosting** | | | | | | | | |
| Precision | 0.737 | 0.025 | 0.765 | 0.028 | 0.782 | 0.034 | 0.778 | 0.031 |
| Recall | 0.745 | 0.025 | 0.668 | 0.023 | 0.670 | 0.025 | 0.667 | 0.024 |
| F1 | 0.737 | 0.025 | **0.700** | 0.024 | **0.705** | 0.027 | **0.703** | 0.024 |
| **Logistic Regression** | | | | | | | | |
| Precision | 0.771 | 0.031 | 0.674 | 0.247 | 0.694 | 0.249 | 0.657 | 0.237 |
| Recall | 0.645 | 0.043 | 0.503 | 0.003 | 0.503 | 0.003 | 0.503 | 0.004 |
| F1 | 0.634 | 0.061 | 0.471 | 0.008 | 0.476 | 0.008 | 0.480 | 0.010 |
| **Naïve Bayes** | | | | | | | | |
| Precision | 0.604 | 0.030 | 0.479 | 0.012 | 0.475 | 0.010 | 0.473 | 0.009 |
| Recall | 0.606 | 0.031 | 0.454 | 0.026 | 0.440 | 0.022 | 0.426 | 0.023 |
| F1 | 0.603 | 0.030 | 0.397 | 0.015 | 0.389 | 0.013 | 0.384 | 0.012 |
| **Neural Network** | | | | | | | | |
| Precision | 0.740 | 0.028 | 0.635 | 0.032 | 0.626 | 0.036 | 0.643 | 0.047 |
| Recall | 0.700 | 0.032 | 0.568 | 0.016 | 0.553 | 0.018 | 0.544 | 0.017 |
| F1 | 0.705 | 0.034 | 0.581 | 0.021 | 0.563 | 0.024 | 0.554 | 0.026 |
| **Random Forest** | | | | | | | | |
| Precision | 0.765 | 0.024 | 0.828 | 0.042 | 0.858 | 0.042 | 0.886 | 0.040 |
| Recall | 0.778 | 0.024 | 0.611 | 0.019 | 0.592 | 0.017 | 0.581 | 0.018 |
| F1 | 0.765 | 0.025 | 0.647 | 0.025 | 0.626 | 0.025 | 0.615 | 0.027 |
| **Support Vector Machine** | | | | | | | | |
| Precision | 0.782 | 0.025 | 0.691 | 0.022 | 0.687 | 0.020 | 0.695 | 0.023 |
| Recall | 0.796 | 0.026 | 0.702 | 0.024 | 0.692 | 0.024 | 0.691 | 0.023 |
| F1 | **0.777** | 0.028 | 0.696 | 0.022 | 0.688 | 0.019 | 0.692 | 0.020 |

*Note:* SD = standard deviation. Numbers highlighted in bold represent the highest $f$-measure. The results are based on 100 experiments.

Table F2: Percentage of Friendly, Helpful, Useful, and Informative Comments

| | *PercFriendComments* | *PercHelpComments* | *PercUseComments* | *PercInfoComments* |
|---|---|---|---|---|
| | (1) | (2) | (3) | (4) |
| *Newcomer* | 0.011 | 0.006 | 0.011*** | 0.008** |
| | (0.007) | (0.004) | (0.004) | (0.004) |
| *Newcomer*×*After* | 0.000 | 0.000 | −0.004 | −0.002 |
| | (0.008) | (0.004) | (0.004) | (0.004) |
| *Tenure* | 0.000*** | 0.000** | 0.000** | 0.000** |
| | (0.000) | (0.000) | (0.000) | (0.000) |
| log(1+*DescLen*) | 0.013*** | 0.016*** | 0.014*** | 0.013*** |
| | (0.002) | (0.001) | (0.001) | (0.001) |
| *LocalDeal* | −0.008** | −0.021*** | −0.019*** | −0.017*** |
| | (0.004) | (0.002) | (0.002) | (0.002) |
| *NumCategories* | 0.003*** | −0.001*** | −0.001*** | −0.001*** |
| | (0.001) | (0.000) | (0.000) | (0.000) |
| *Content* | 0.074*** | −0.003 | 0.005* | 0.003 |
| | (0.006) | (0.003) | (0.003) | (0.003) |
| Time FE (Day, Hour) | Yes | Yes | Yes | Yes |
| Category FE | Yes | Yes | Yes | Yes |
| Observations | 39,197 | 39,197 | 39,197 | 39,197 |
| Adjusted $R$-squared | 0.018 | 0.055 | 0.047 | 0.050 |

*Note:* FE = fixed effects. Robust standard errors clustered by user are in parentheses. *** $p$ <0.01, ** $p$ <0.05, * $p$ <0.10.

## References

Arslan, R. C., Walther, M. P., & Tata, C. S. (2020). formr: a study framework allowing for automated feedback generation and complex longitudinal experience-sampling studies using R. *Behavior Research Methods*, *52*(1), 376–387.

Clarke, J., Chen, H., Du, D., & Hu, Y. J. (2020). Fake news, investor attention, and market reaction. *Information Systems Research*, *32*(1), 35–52.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284.

Liu, X., Wang, G. A., Fan, W., & Zhang, Z. (2020). Finding useful solutions in online knowledge communities: A theory-driven design and multilevel analysis. *Information Systems Research*, *31*(3), 731–752.

Natterson-Horowitz, B., & Bowers, K. (2020). *Wildhood: The astounding connections between human and animal adolescents*. Scribner.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Punyon, J., & Montrose, K. (2020). *The unfriendly robot: Automatically flagging unwelcoming comments*. Retrieved April 1, 2024, from https://stackoverflow.blog/2020/04/09/the-unfriendly-robot-automatically-flagging-unwelcoming-comments/

Remus, R., Quasthoff, U., & Heyer, G. (2010). SentiWS - a publicly available German-language resource for sentiment analysis. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, 1168–1171.

Shen, W., Hu, Y. J., & Ulmer, J. R. (2015). Competing for attention: An empirical study of online reviewers' strategic behavior. *MIS Quarterly*, *39*(3), 683–696.

Sidarenka, U., & Stede, M. (2016). Generating sentiment lexicons for German Twitter. *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, 80–90.

Wenninger, H., Krasnova, H., & Buxmann, P. (2019). Understanding the role of social networking sites in the subjective well-being of users: A diary study. *European Journal of Information Systems*, *28*(2), 126–148.

Yin, D., Bond, S. D., & Zhang, H. (2014). Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews. *MIS Quarterly*, *38*(2), 539–560.