Email Campaign Click Through Rate Prediction with Linear Regression

Shang Chen
srochen@ucdavis.edu

Junwon Choi
jnwchoi@ucdavis.edu

Hangyu Li
alnli@ucdavis.edu

Abstract

This report details the construction of a linear regression model to predict customer click-through rates (CTR) in email campaigns, aiming to optimize model efficiency by minimizing data requirements while preserving predictive performance. Using a dataset from the Analytics Vidhya Job-a-thon, we streamlined the model by selecting relevant variables, addressing multicollinearity, and assessing variable significance. The final model balances bias and accuracy, providing insights that can guide the development of cost-effective and high-performing email marketing strategies.

1 Introduction

Despite the prevalence of social media marketing, email campaigns remain a popular and effective strategy in business and marketing [3]. The effectiveness of these campaigns is primarily attributed to their highly targeted nature. However, running email campaigns—particularly for small businesses—can be costly, especially when they leverage granular customer data and personalized content [2].

The primary motivation for this project is to develop a linear regression model capable of predicting customer click-through rate (CTR) - labeled as *click_rate* - while minimizing the amount of information required for its construction. This approach aims to reduce costs for businesses without compromising predictive performance.

For this study, we utilized email marketing data [1] from the Analytics Vidhya Job-a-thon held in August 2022, an Indian data science competition. The dataset consists of 1,888 cases with 22 variables and no missing values. The *campaign_id* variable was dropped as it was an identifier (non explanatory) variable and the *is_timer* variable was dropped as it only contained 1 unique value. Hence 20 total variables were used. The original dataset is publicly available on Kaggle, and more details about the dataset are outlined in Appendix A. We plan to answer the following questions through our research:

1. Are there any critical outliers in the dataset influencing downstream analysis?
2. What do the variable and residual distributions look like?
3. Which variables are influential in determining whether or not customers will click on at least one campaign email?
4. Among all variables, which have positive or negative relationships with CTR? Which variables display no particular relationship with CTR?
5. How well does a baseline model (1st order full additive model) perform? Are there any non-significant variables present in the full baseline model?
6. Are there any motivating patterns that would require the use of additional polynomial and/or interaction terms?
7. How concise can a model be without negatively impacting overall performance (minimizing bias, high accuracy, etc.)?

2 Methods and Results

2.1 Exploratory Data Analysis and Cleaning

As previously noted, we began by dropping the two irrelevant variables, *campaign_id* and *is_timer*, before conducting any analysis. Next, we ensured that all variables were assigned the correct data types (e.g., categorical variables were set as `factor` in R). The dataset was then split into a training set of $n = 1,479$ (80%) and a validation set of $n = 370$ (20%). We did not use the separate test data provided in the Kaggle competition because the *click_rate* column was undisclosed. During our initial exploratory data analysis (EDA), we encountered a significant challenge involving concerning

patterns in the dataset. Figure 1 shows diagnos-

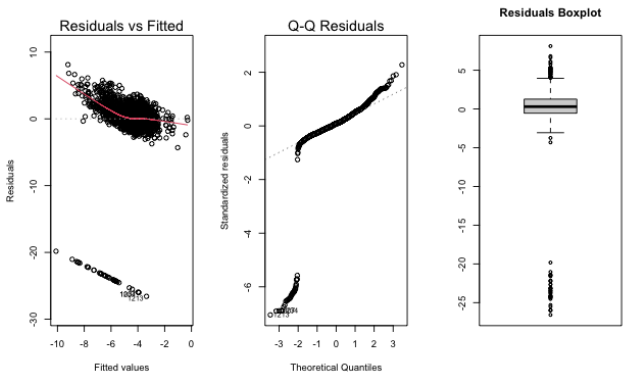

Figure 1: Diagnostic Plots of First Order Additive Model (including *click_rate = 0*)

tic plots for a simple first-order additive model that includes all variables (**Model-0**). The clustered outliers correspond to cases where *click_rate* equals 0, accounting for 2.1% of all observations. To investigate further, we temporarily removed these cases and repeated the EDA (Figure 3). To

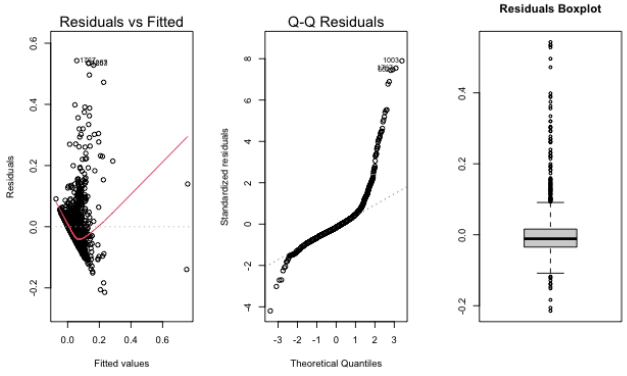

Figure 2: Diagnostic Plots of First Order Additive Model (excluding *click_rate = 0*)

address the issues of an increasing spread of residuals with fitted values and the heavy-tailed Q-Q plot, we applied a Box-Cox transformation ($\lambda = 0$), which involves taking the log of *click_rate*. Figure 4 displays the histograms of *click_rate*, before and after the transformation, which appears to be normally distributed. To further examine the effect of the transformation, side-by-side boxplots were created to compare the distributions

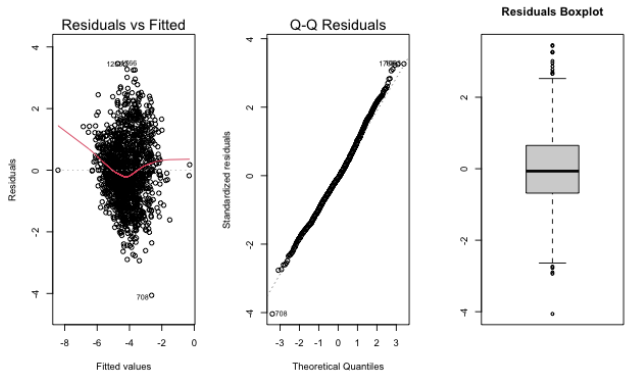

Figure 3: Diagnostic Plots of First Order Additive Model (excluding *click_rate = 0*) with $\log(click\_rate)$

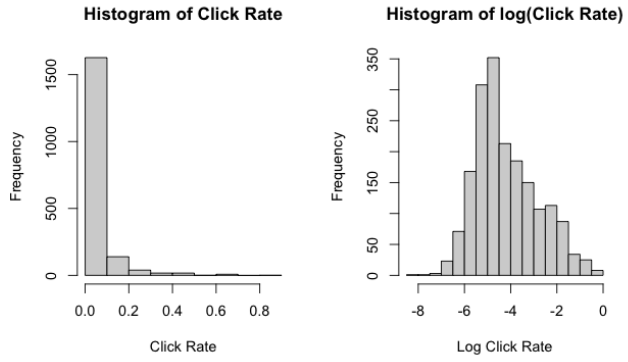

Figure 4: Histograms of $click\_rate$ **and** $\log(click\_rate)$

of *log(click_rate)* across different classes within categorical variables. Additional boxplots are provided in Appendix B.1. Boxplots in Figure 5 show

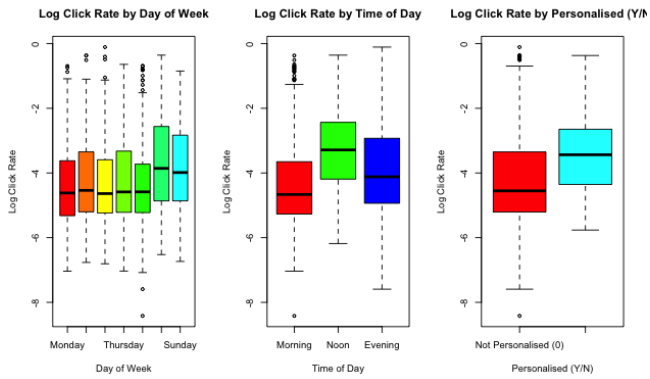

Figure 5: Side-by-side Boxplots of *log(click_rate)* by *day_of_week, time_of_day, is_personalised*

that the distribution of $\log(click\_rate)$ are similar among classes for variables such as *day_of_week, time_of_day, is_personalised*, while boxplots of variables consisting of more number of categories such as *category* (Figure 6, *sender, target_audience* (Appendix B.1 display much more variability in distributions among classes. Histograms of the quantitative variables showed that while variables such as *subject_length, body_length, mean_paragraph_length, mean_cta_length* appear to be roughly normally distributed, other quantitative variables with a more restricted range of values were not.

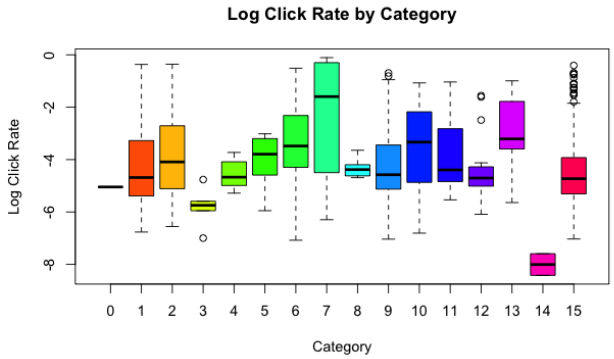

Figure 6: Side-by-side Boxplots of *log(click_rate)* by *category*

Hence the skeleton model to begin further downstream analysis had a log-transformed response variable with all 19 first order additive terms (**Model-1**).

2.2 Dealing with Multicollinearity

We identified NAs in the R output for the summary of **Model-1**, likely due to high multicollinearity. This suspicion was further confirmed when attempting to calculate the *VIF* for **Model-1**, which resulted in an error when using the `vif()` function in R. The variables of concern were *day_of_week* and *product*. After removing these two variables, we formulated **Model-2**.

By examining the *VIF* of **Model-2** (Table 4), we observed that *sender* and *category* have *VIF* values exceeding 10, with values of 28.24 and 45.75, respectively, indicating strong multicollinearity with other variables. We therefore reduced the model

Variable	$GVIF$	Df	$GVIF^{1/(2\cdot Df)}$
sender	28.238	11	1.164
subject_len	1.547	1	1.244
body_len	2.786	1	1.669
mean_paragraph_len	1.963	1	1.401
is_weekend	1.181	1	1.087
times_of_day	1.344	2	1.077
category	**45.746**	14	1.146
no_of_CTA	2.464	1	1.570
mean_CTA_len	1.260	1	1.123
is_image	1.114	1	1.055
is_personalised	1.250	1	1.118
is_quote	1.144	1	1.070
is_emoticons	1.096	1	1.047
is_discount	1.103	1	1.050
is_price	1.064	1	1.032
is_urgency	1.189	1	1.091
target_audience	4.302	16	1.047

Table 1: VIF values from Model-2

to exclude *category* that had the highest VIF value, which we set as **Model-3**. Removing this variable resolved the multicollinearity issue, and the *VIF* values for **Model-3** fell within the range of 1 to 3 for all variables. The VIF values from **Model-3** can be found in Appendix C.

2.3 Baseline Model & Model Building Methodology

Throughout the project, we adhered to a strict and consistent significance level of $\alpha = 0.05$ for both t-tests and F-tests. We observed that the *p-values* from the t-tests for certain terms in the dummy variables *sender* and *target_audience* in **Model-3** exceeded 0.05. Considering that these two variables might have a business-related relationship, we conducted further investigation using F-tests. Specifically, we compared models with and without *sender* and *target_audience* as reduced models.

The results indicated that these terms could not be excluded from **Model-3**, whether individually or collectively. Consequently, we selected **Model-3** as the baseline model for our analysis. The equation for **Model-3** can be found in Appendix D.

Building on this baseline model, our model development process followed the structure illustrated

in Figure 7. The left branch emphasizes reducing a first-order additive model, while the right branch focuses on constructing a higher-order interaction model to capture more nuanced patterns in the data. All stepwise regression techniques used in our project utilized forward, backward, and forward-backward procedures. All stepwise regression techniques used in our project utilized backward and forward-backward procedures from the "full" model. Interestingly, both approaches yielded identical results when applied at various stages of the project.

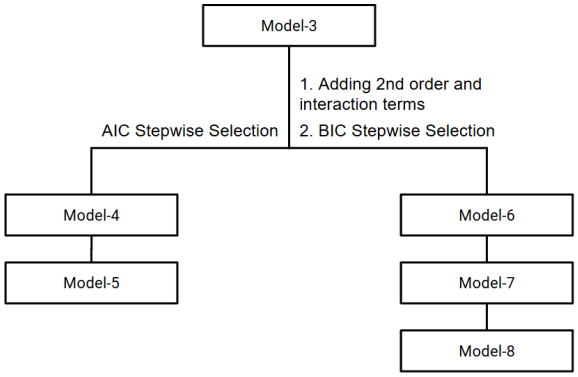

Figure 7: Model Building Workflow

2.4 First Order Additive Model Construction (Left branch)

To avoid including nuisance variables in our model, we conducted stepwise selection starting with **Model-3**. Since it offers higher predictive power and is more conservative, we selected AIC as our criterion. Meanwhile, we accounted for interaction and polynomial terms in the right branch, where selection using BIC was performed later. The selection process concluded with **Model-4**, which retained only the variables chosen by stepwise AIC.

However, stepwise selection primarily focuses on minimizing the sum of squares error, which evaluates the model's overall performance. Upon examining the R output of **Model-4**, we found that *no_of_CTA* and *is_image* were non-significant according to their *p-values*. To further evaluate whether these variables should be retained, we conducted three F-tests to assess the effect of dropping them individually and together:

The results showed that only the p-value for dropping *no_of_CTA* was below 0.05. Consequently, we removed *no_of_CTA* and finalized our model (**Model-5**) for the left branch. The equation for **Model-5** can be found in Appenndix D.

2.5 Second Order Model with Interaction Construction (Right branch)

With many terms added to the model (**Model-6**), we chose BIC as the criterion to impose more penalties on the complex model. The selection process resulted in **Model-7**. Eight first-order terms (*sender*, *mean_paragraph_len*, *is_weekend*, *mean_CTA_len*, *is_image*, *is_personalised*, *is_quote*, *is_price*) and their corresponding interaction terms were dropped by the BIC selection. No interaction terms were retained, and only two quadratic terms were found to be significant: the second-order terms of *body_len* and *no_of_CTA*. The full **Model-7** performed well in the F-test, with a p-value < 0.05. We marked it as one of the final results of the right branch.

Model 7:

$$
\begin{aligned}
\log(\hat{click\_rate}) &= -3.836 - 0.003082 \cdot subject\_len \\
&\quad - 0.0001084 \cdot body\_len \\
&\quad + 0.6255 \cdot times\_of\_day_{Morning} \\
&\quad + 0.4162 \cdot times\_of\_day_{Noon} \\
&\quad - 0.06508 \cdot no\_of\_CTA \\
&\quad - 0.1468 \cdot is\_emoticons - 0.8203 \cdot is\_discount_Y \\
&\quad - 0.4103 \cdot is\_urgency_Y + 0.953 \cdot ta_1 + 0.196 \cdot ta_2 \\
&\quad + 1.455 \cdot ta_3 + 0.5341 \cdot ta_4 + 0.6108 \cdot ta_5 \\
&\quad + 1.326 \cdot ta_6 + 0.6799 \cdot ta_7 - 0.1935 \cdot ta_8 \\
&\quad + 0.7469 \cdot ta_9 + 0.994 \cdot ta_{10} - 1.825 \cdot ta_{11} \\
&\quad + 1.157 \cdot ta_{12} + 0.9207 \cdot ta_{13} + 1.991 \cdot ta_{14} \\
&\quad + 0.1299 \cdot ta_{15} + 0.2079 \cdot ta_{16} \\
&\quad + 2.295 \times 10^{-9} \cdot body\_len^2 \\
&\quad + 0.001399 \cdot no\_of\_CTA^2
\end{aligned}
$$

Here, ta_1 - ta_{16} represents $target\_audience_1$ - $target\_audience_{16}$. In **Model-3**, we identified four significant variables: *sender*, *is_weekend*, *is_personalised*, and *is_price*, but these were later dropped by the BIC selection. To ensure we did not overlook significant variables, we manually added them back, resulting in **Model-8**. Comparing **Model-8** with **Model-7** using the F-test, the new **Model-8** was retained as another candidate in our right branch.

Model 8:

$$
\begin{aligned}
\log(\hat{click\_rate}) &= -3.737 - 0.002919 \cdot subject\_len \\
&- 0.00009598 \cdot body\_len \\
&+ 0.5728 \cdot times\_of\_day_{Morning} \\
&+ 0.3995 \cdot times\_of\_day_{Noon} \\
&- 0.0706 \cdot no\_of\_CTA \\
&- 0.1528 \cdot is\_emoticons - 0.7857 \cdot is\_discount_Y \\
&- 0.4317 \cdot is\_urgency_Y + 0.8303 \cdot ta_1 + 0.4691 \cdot ta_2 \\
&+ 1.365 \cdot ta_3 + 0.5374 \cdot ta_4 + 0.5862 \cdot ta_5 \\
&+ 1.352 \cdot ta_6 + 0.6877 \cdot ta_7 + 0.1815 \cdot ta_8 \\
&+ 0.6463 \cdot ta_9 + 0.952 \cdot ta_{10} - 1.846 \cdot ta_{11} \\
&+ 1.133 \cdot ta_{12} + 0.9224 \cdot ta_{13} + 1.951 \cdot ta_{14} \\
&+ 0.1094 \cdot ta_{15} + 0.1724 \cdot ta_{16} \\
&+ 2.063 \times 10^{-9} \cdot body\_len^2 \\
&+ 0.001615 \cdot no\_of\_CTA^2 - 1.872 \cdot sender_1 \\
&+ 0.07868 \cdot sender_2 - 0.205 \cdot sender_3 \\
&- 0.3381 \cdot sender_6 + 2.374 \cdot sender_7 \\
&- 0.647 \cdot sender_9 - 0.4813 \cdot sender_{10} \\
&+ 0.7673 \cdot sender_{11} - 0.8566 \cdot sender_{12} \\
&- 1.564 \cdot sender_{14} - 0.00676 \cdot sender_{15} \\
&+ 0.1703 \cdot is\_weekend_Y \\
&+ 0.3131 \cdot is\_personalised_Y \\
&- 0.0001239 \cdot is\_price
\end{aligned}
$$

2.6 Model Validation and Selection

Table 2 presents the internal and external validation of our candidate models based on various criteria after addressing outliers (explained further in Section 2.7. We set the **Model-3** as the baseline model and included the three candidate Models (**Model-5**, **Model-7**, **Model-8**).

Comparing R^2 and R^2_a, we found that **Model-8** (R^2=0.333 and R^2_a=0.316) has the best ability of explaining the linear relation between click rate and X variables. Meanwhile, it also has a high complexity (p=39) and lowest Mallows' C_p which demonstrates a reduced propensity for overfitting. Another well-performed model would be **Model-7** (R^2=0.311 and R^2_a=0.299), which has fewest variables. MSE addresses the predictive power of the model on the training data and **Model-7** has the best result(MSE=0.00557) showing its strength in predicting. After removing the leverage values, all models have an accessible $Press_p$. **Model-8** has the smallest one among them. AIC and BIC are addressing the similar idea, **Model-8** reaches the lowsest AIC (=366.111) and **Model-7** has the lowest BIC (=533.950), indicating its balance between goodness-of-fit and model simplification.

For model selection, we chose **Model-7** and **Model-8** as our final two models. The low MSE of **Model-7** enhances its predictive accuracy, while its lowest BIC and smallest model size among the four candidate models contribute to a simplified yet effective design. Additionally, **Model-7** maintains moderate R^2 and R^2_a, balancing simplicity and performance.

Although the MSE of **Model-8** is the second lowest among the four models, it achieves the best performance in R^2 and R^2_a. This reflects a trade-off between explanatory power and predictive accuracy, which makes **Model-8** a compelling alternative.

2.7 Outliers

We applied Bonferroni's Threshold to identify outliers. However, the changes on model fitting performance are subtle in terms of validation criteria and coefficients. This implies that these outliers are not influential and our models are robust. Cook's distance values for **Model-7** are plotted in Figure 8, and all other distance plots can be found in Appendix E.

Table 2: Candidate Model Validation

Model	R^2	R_a^2	C_p	p	AIC_p	BIC_p	$Press_p$	MSE
Model 3	0.316	0.296	192.177	41	409.381	626.645	1948.025	0.00619
Model 5	0.312	0.295	192.135	37	408.788	604.855	1945.212	0.00613
Model 7	0.311	0.299	172.383	27	390.873	**533.950**	1925.105	**0.00557**
Model 8	**0.333**	**0.316**	**146.641**	39	**366.111**	572.777	**1890.217**	0.00587

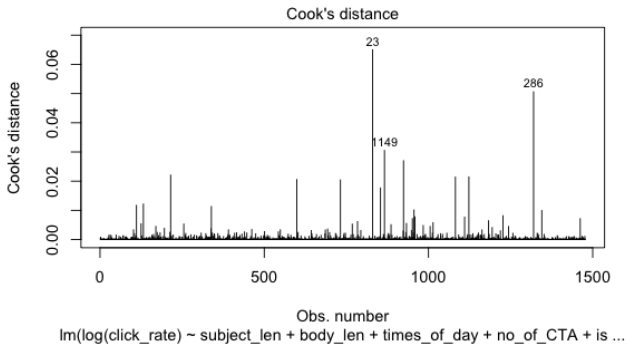

Figure 8: Cook's Distance Plot for Model-7

3 Conclusion and Discussion

Examining the two final models provides valuable insights for email campaigns. Since a log transformation was applied to the response variable *click_rate*, variables with negative coefficients can be viewed as passive factors. What aligns with people's common sense is the length of subject and body of the email does affect the click rate - the shorter, the better. But other variables that capture customer's attention and interest were shown to also be valuable in predicting click behavior. Sending personalized emails on weekends increases click rates, likely because recipients have more free time and find personalized content more engaging. Surprisingly, emails sent in the morning outperform those sent in the afternoon or evening, and adding emoticons or promotional deals does not significantly boost clicks.

The most important takeaway for marketing is the importance of targeting specific audience groups. Both **Model-7** and **Model-8** show that the 10th audience group has a negative coefficient, suggesting they are less responsive, while the 3rd, 6th, 12th, and 13th groups exhibit similar positive preferences. Marketers should tailor email content to better engage these responsive groups while exploring strategies to re-engage the 10th group to strengthen customer relationships and drive product promotions effectively.

4 Limitations and Future Work

An interesting direction for future work would be to develop separate models for each category within the categorical variables. The current approach in this project aims to identify key features for predicting consumers' click rates in a general context by using dummy variables to represent categorical predictors. However, in a real-world marketing setting, it might be more beneficial to extract features tailored to specific categories, especially when certain information about the target audience is already available. While this approach could offer more precise insights, it presents challenges. Managing a large number of models would significantly increase the complexity of the project, and achieving reliable results would require a substantially larger sample size. These practical considerations must be weighed against the potential benefits of such an approach in real-world applications. A binary classification task (beyond the scope of STA 206) to identify customers with zero click rates would also be beneficial in a business context, allowing more targeted campaigns to attract non-responding customers.

References

[1] 2024. Email Click-Through Rate (CTR) Prediction Dataset. https://www.kaggle.com/datasets/sk4467/email-ctr-prediction/data Accessed: 2024-12-09.
[2] Abby Fields. 2024. How much does email marketing cost in 2024? https://www.webfx.com/email-marketing/pricing/
[3] Mike Maynard. 2024. Council post: Email marketing still works-and it's more effective than ever. https://www.forbes.com/councils/forbesagencycouncil/2024/06/21/email-marketing-still-works-and-its-more-effective-than-ever/

5 Appendix

A Dataset Details

A.1 Variables & Descriptions

Table 3: Email CTR Data Variable Descriptions

Variable	Description	Type
click_rate	Click through rate (CTR)	Numeric (**Response Variable**)
campaign_id	Unique identifier of an email campaign	Numeric (Not Used)
sender	Sender of email	Categorical
subject_len	Length of email subject	Numeric
body_len	Length of the body(in character)	Numeric
mean_paragraph_len	Average length of the body(in character)	Numeric
day_of_week	Day of the week email is sent	Categorical (from 0: Monday to 6: Sunday)
is_weekend	Whether email is sent on weekends	Categorical (0: No 1: Yes)
times_of_day	Time of the day email was sent	Categorical (Morning/Noon/Evening)
category	Category of the product included in email	Categorical (from "0" to "15")
product	Product number included in email	Categorical (from "0" to "43")
no_of_CTA	Number of CTA in email	Numeric
mean_CTA_len	Average length of the CTA	Numeric
is_image	Number of images in email	Numeric
is_personalised	Whether email is personalized	Categorical (0: No 1: Yes)
is_quote	Number of quotes in the email	Numeric
is_timer	Whether email contains a timer	Categorical (Not Used)
is_emoticons	Number of emoticons in email	Numeric
is_discount	Whether email offers discount	Categorical (0: No 1: Yes)
is_price	Whether email contains a price	Categorical (0: No 1: Yes)
is_urgency	Whether email entails urgency	Categorical (0: No 1: Yes)
target_audience	Cluster label of target audience	Categorical (from "0" to "16")

B EDA Visualizations

B.1 Side-by-side Boxplots for Categorical Variables

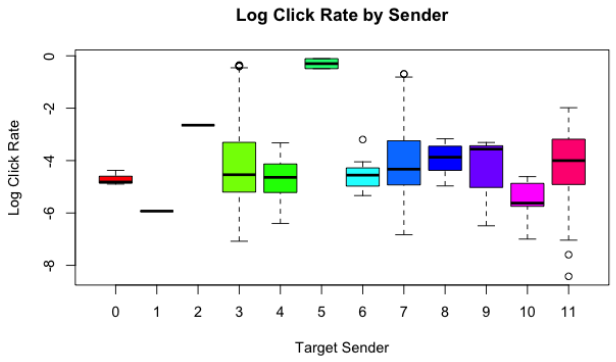

Figure 9: Side-by-side Boxplots of *log(click_rate)* by *sender*

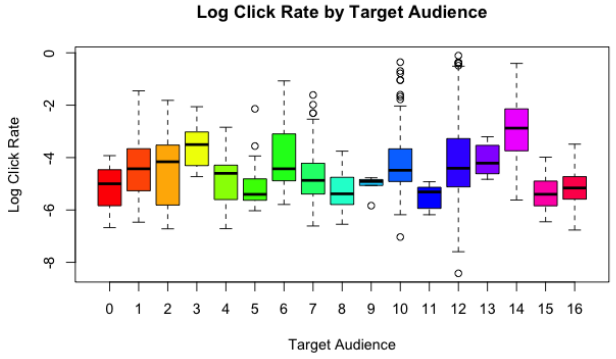

Figure 10: Side-by-side Boxplots of *log(click_rate)* by *target_audience*

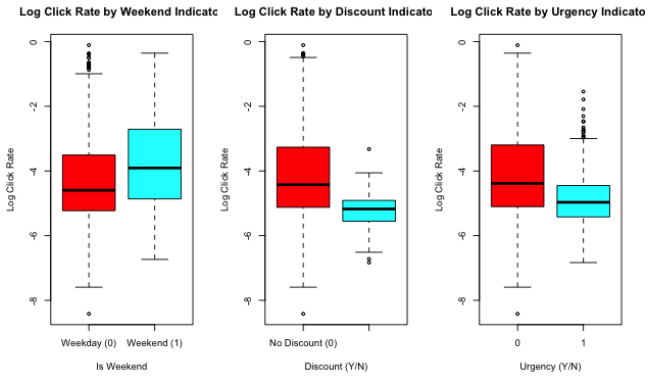

Figure 11: Side-by-side Boxplots of *log(click_rate)* by *is_weekend, is_discount, is_urgency*

B.2 Histogram of Quantitative Variables

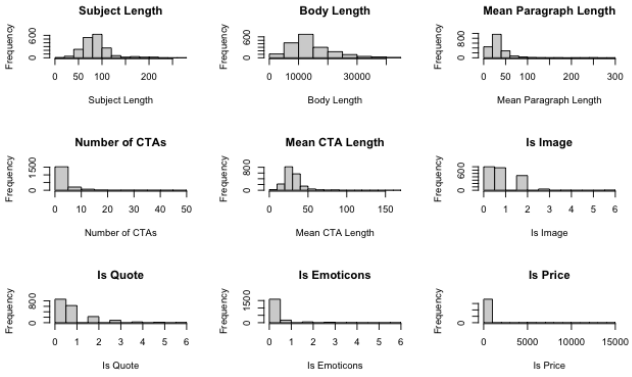

Figure 12: Histogram of All Quantitative Variables

C VIF Values

Variable	$GVIF$	Df	$GVIF^{1/(2\cdot Df)}$
sender	3.144	11	1.053
subject_len	1.499	1	1.224
body_len	2.645	1	1.626
mean_paragraph_len	1.817	1	1.348
is_weekend	1.165	1	1.079
times_of_day	1.258	2	1.059
no_of_CTA	2.036	1	1.427
mean_CTA_len	1.220	1	1.105
is_image	1.052	1	1.026
is_personalised	1.215	1	1.102
is_quote	1.073	1	1.036
is_emoticons	1.074	1	1.037
is_discount	1.073	1	1.036
is_price	1.060	1	1.029
is_urgency	1.111	1	1.054
target_audience	2.102	16	1.023

Table 4: VIF values from Model-3

D Model Equations

Model 3:

$$\begin{aligned}\log(\hat{click\_rate}) &= -4.117 - 2.324 \cdot sender_1 \\ &+ 0.110 \cdot sender_2 - 0.3522 \cdot sender_3 \\ &- 0.4238 \cdot sender_6 + 2.920 \cdot sender_7 \\ &- 0.2466 \cdot sender_9 - 0.5905 \cdot sender_{10} \\ &+ 0.5881 \cdot sender_{11} - 0.9683 \cdot sender_{12} \\ &- 1.731 \cdot sender_{14} - 0.08321 \cdot sender_{15} \\ &- 0.002460 \cdot subject\_len - 0.00002408 \cdot body\_len \\ &+ 0.0004930 \cdot mean\_paragraph\_len \\ &+ 0.1728 \cdot is\_weekend_Y \\ &+ 0.6252 \cdot times\_of\_day_{Morning} \\ &+ 0.4036 \cdot times\_of\_day_{Noon} \\ &- 0.01375 \cdot no\_of\_cta \\ &- 0.005462 \cdot mean\_CTA\_len \\ &- 0.06546 \cdot is\_image \\ &+ 0.4879 \cdot is\_personalised_Y \\ &- 0.03311 \cdot is\_quote \\ &- 0.1680 \cdot is\_emoticons - 0.7586 \cdot is\_discount_Y \\ &- 0.0001129 \cdot is\_price - 0.4487 \cdot is\_urgency_Y \\ &+ 0.8283 \cdot ta_1 + 0.7016 \cdot ta_2 + 1.441 \cdot ta_3 \\ &+ 0.4080 \cdot ta_4 + 0.5006 \cdot ta_5 + 1.333 \cdot ta_6 \\ &+ 0.7085 \cdot ta_7 - 0.03886 \cdot ta_8 + 0.6472 \cdot ta_9 \\ &+ 1.029 \cdot ta_{10} - 0.2891 \cdot ta_{11} + 1.163 \cdot ta_{12} \\ &+ 0.8390 \cdot ta_{13} + 2.051 \cdot ta_{14} + 0.05438 \cdot ta_{15} \\ &+ 0.1305 \cdot ta_{16}\end{aligned}$$

Model 5:

$$\begin{aligned}\log(\hat{click\_rate}) &= -4.095 - 2.281 \cdot sender_1 \\ &+ 0.1895 \cdot sender_2 - 0.03655 \cdot sender_3 \\ &- 0.4126 \cdot sender_6 + 2.933 \cdot sender_7 \\ &- 0.4634 \cdot sender_9 - 0.5644 \cdot sender_{10} \\ &+ 0.6261 \cdot sender_{11} - 0.9790 \cdot sender_{12} \\ &- 1.674 \cdot sender_{14} - 0.09112 \cdot sender_{15} \\ &- 0.002889 \cdot subject\_len - 0.00002979 \cdot body\_len \\ &+ 0.1649 \cdot is\_weekend_Y \\ &+ 0.6215 \cdot times\_of\_day_{Morning} \\ &+ 0.4045 \cdot times\_of\_day_{Noon} \\ &- 0.00559 \cdot mean\_CTA\_len \\ &+ 0.4785 \cdot is\_personalised_Y \\ &- 0.1661 \cdot is\_emoticons - 0.7642 \cdot is\_discount_Y \\ &- 0.0001058 \cdot is\_price - 0.4354 \cdot is\_urgency_Y \\ &+ 0.8518 \cdot ta_1 + 0.7092 \cdot ta_2 + 1.447 \cdot ta_3 \\ &+ 0.3283 \cdot ta_4 + 0.4444 \cdot ta_5 + 1.328 \cdot ta_6 \\ &+ 0.6796 \cdot ta_7 - 0.01598 \cdot ta_8 + 0.6377 \cdot ta_9 \\ &+ 1.027 \cdot ta_{10} - 0.2964 \cdot ta_{11} + 1.142 \cdot ta_{12} \\ &+ 0.7984 \cdot ta_{13} + 2.038 \cdot ta_{14} + 0.04929 \cdot ta_{15} \\ &+ 0.0982 \cdot ta_{16}\end{aligned}$$

E Cook's Distance Plots

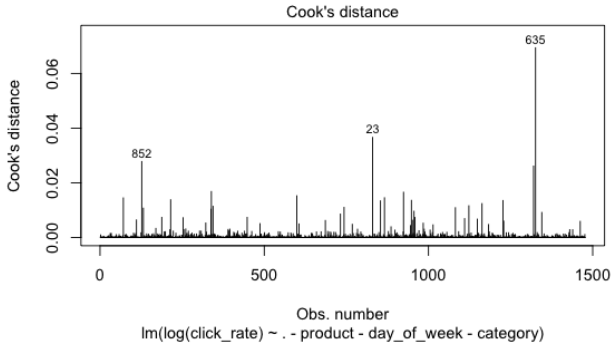

Figure 13: Cook's Distance Plot for Model-3

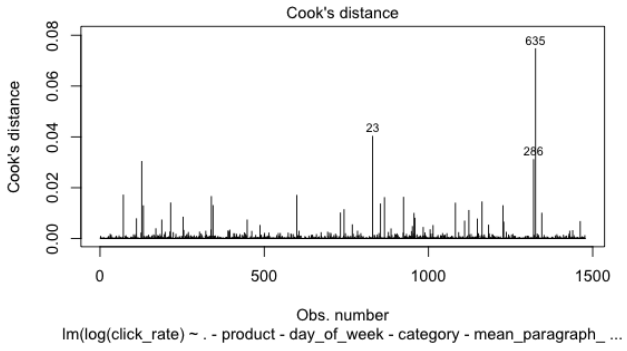

Figure 14: Cook's Distance Plot for Model-5

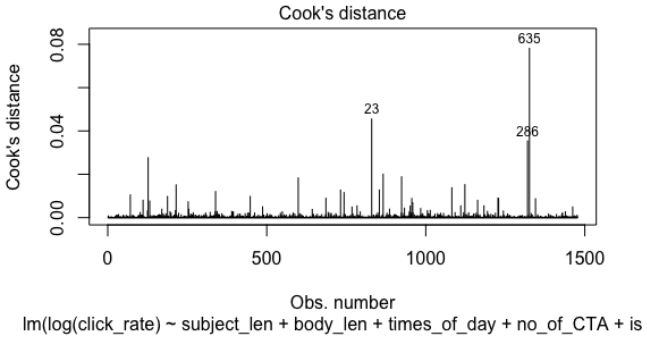

Figure 15: Cook's Distance Plot for Model-8

STA 206 Final Project Code

STA 206 | Fall 2024

Dec. 9, 2024

Data Import, Cleaning, EDA

```
email <- read.csv('email/train_data.csv')
```

```
# Check for number of missing values by column
missing_values <- colSums(is.na(email))
print(missing_values)
```

```
##        campaign_id             sender        subject_len           body_len
##                  0                  0                  0                  0
## mean_paragraph_len        day_of_week         is_weekend       times_of_day
##                  0                  0                  0                  0
##           category            product          no_of_CTA       mean_CTA_len
##                  0                  0                  0                  0
##           is_image    is_personalised           is_quote           is_timer
##                  0                  0                  0                  0
##       is_emoticons        is_discount           is_price         is_urgency
##                  0                  0                  0                  0
##    target_audience         click_rate
##                  0                  0
```

```
# Change categorical variables to factors
email$times_of_day <- as.factor(email$times_of_day)
email$sender <- as.factor(email$sender)
email$category <- as.factor(email$category)
email$product <- as.factor(email$product)
email$target_audience <- as.factor(email$target_audience)
email$day_of_week <- as.factor(email$day_of_week)
email$is_personalised <- as.factor(email$is_personalised)
email$is_discount <- as.factor(email$is_discount)
email$is_urgency <- as.factor(email$is_urgency)
email$is_weekend <- as.factor(email$is_weekend)

# number of unique values in each column
sapply(email, function(x) length(unique(x)))
```

```
##        campaign_id             sender        subject_len           body_len
##               1888                 12                170               1568
## mean_paragraph_len        day_of_week         is_weekend       times_of_day
##                121                  7                  2                  3
##           category            product          no_of_CTA       mean_CTA_len
##                 16                 43                 40                 79
##           is_image    is_personalised           is_quote           is_timer
```

```
##                 6                 2                 7                 1
##       is_emoticons       is_discount          is_price       is_urgency
##                 6                 2                 8                 2
##    target_audience        click_rate
##                17              1810
```

```
# remove `is_timer` from the dataset
email <- email[, -which(names(email) == "is_timer")]

# Delete `campaign_id` column
email <- email[, -1]

# Subset based on `email$click_rate` not equal to 0
email_yes_click <- email[email$click_rate != 0, ]

# Keep full dataset for later use
email_full <- email
```

```
str(email)
```

```
## 'data.frame':    1888 obs. of  20 variables:
##  $ sender            : Factor w/ 12 levels "0","1","2","3",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ subject_len       : int  76 54 59 74 80 54 54 60 89 89 ...
##  $ body_len          : int  10439 2570 12801 11037 10011 2569 2570 12117 10055 11049 ...
##  $ mean_paragraph_len: int  39 256 16 30 27 256 256 17 14 26 ...
##  $ day_of_week       : Factor w/ 7 levels "0","1","2","3",..: 6 6 6 5 6 5 5 5 5 4 ...
##  $ is_weekend        : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 1 1 1 ...
##  $ times_of_day      : Factor w/ 3 levels "Evening","Morning",..: 3 2 3 1 3 1 1 3 3 1 ...
##  $ category          : Factor w/ 16 levels "0","1","2","3",..: 7 3 3 16 7 3 3 7 7 16 ...
##  $ product           : Factor w/ 43 levels "0","1","2","3",..: 27 12 12 10 27 12 12 27 27 10 ...
##  $ no_of_CTA         : int  3 0 3 4 3 0 0 4 3 4 ...
##  $ mean_CTA_len      : int  29 22 23 24 31 22 22 34 34 28 ...
##  $ is_image          : int  0 0 1 0 0 0 0 1 1 0 ...
##  $ is_personalised   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_quote          : int  0 0 1 0 1 0 0 1 0 0 ...
##  $ is_emoticons      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ is_discount       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ is_price          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ is_urgency        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ target_audience   : Factor w/ 17 levels "0","1","2","3",..: 15 11 17 11 15 11 11 15 14 7 ...
##  $ click_rate        : num  0.10308 0.7 0.00277 0.01087 0.14283 ...
```

```
# Illustrating issue with keeping `mail_full$click_rate` == 0

email_full_added_noise <- email_full
email_full_added_noise$click_rate <- email_full$click_rate + 0.0000000000001

model1_added_noise <- lm(click_rate ~ ., data = email_full_added_noise)

par(mfrow=c(1,3))
plot(model1_added_noise, which=1)
plot(model1_added_noise, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   509, 697, 837, 876, 906, 907, 1109, 1162, 1387
```

```r
boxplot(model1_added_noise$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))

library(MASS)
boxcox(model1_added_noise)
```

```r
model2_added_noise <- lm(log(click_rate) ~ ., data = email_full_added_noise)
par(mfrow=c(1,3))
plot(model2_added_noise, which=1)
plot(model2_added_noise, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   509, 697, 837, 876, 906, 907, 1109, 1162, 1387
```

```r
boxplot(model2_added_noise$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))
```

We continue with non-zero `click_rate` data.

Splitting Train/Validation

```r
set.seed(100)
n <- nrow(email_yes_click)
ind <- sample(1:n, 0.8*n, replace=FALSE)
train <- email_yes_click[ind, ] # training set
valid <- email_yes_click[-ind, ] # validation/test set
```

```r
# Subset numeric columns
numeric_cols <- sapply(train, is.numeric)
numeric_data <- train[, numeric_cols]

# Calculate correlation
cor_matrix <- cor(numeric_data)
print(cor_matrix)
```

Initial Full Model & Baseline Model Construction

```r
model0 <- lm(click_rate ~ ., data = train) # full model
summary(model0)

par(mfrow=c(1,3))
plot(model0, which=1)
plot(model0, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   311, 493, 744, 757, 795, 1286, 1345
```

```r
boxplot(model0$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))
```

```r
boxcox(model0)
```

```r
model1 <- lm(log(click_rate) ~ ., data = train)
summary(model1)

par(mfrow=c(1,3))
plot(model1, which=1)
plot(model1, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   311, 493, 744, 757, 795, 1286, 1345
```

```r
boxplot(model1$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))
```

Check for Multicollinearity - before dropping any variables

```r
library(car)
```

```
## Loading required package: carData
```

```r
vif(model1) # has error due to perfect multicollinearity between some variables (we
↪  suspect that `day_of_week` and `product` are causing this due to NA output in the
↪  summary)
```

```
## Error in vif.default(model1): there are aliased coefficients in the model
```

Remove perfectly correlated variables

```r
model2 <- lm(log(click_rate) ~ . -day_of_week - product, data = train)
summary(model2)
```

```r
vif(model2)
```

```r
model3 <- lm(log(click_rate) ~ . -product -day_of_week -category, data = train)
summary(model3)
vif(model3)
```

```r
# model4.1 <- lm(log(click_rate) ~ . -product -day_of_week -category -sender, data =
↪  train)
# model4.2 <- lm(log(click_rate) ~ . -product -day_of_week -category -target_audience,
↪  data = train)
# model4.3 <- lm(log(click_rate) ~ . -product -day_of_week -category -sender
↪  -target_audience, data = train)
#
# anova(model4.1, model3)
# anova(model4.2, model3)
# anova(model4.3, model3)
```

```r
# model4 <- model3
```

```r
library(MASS)

stepAIC(model3, scope=list(upper=model3, lower = ~1), direction="both", k=2, trace =
↪  FALSE)
# stepAIC(model3, scope=list(upper=model3, lower = ~1), direction="backward", k=2, trace
↪  = FALSE)
```

Left Branch

```r
model4 <- lm(log(click_rate) ~ . -product -day_of_week -category  -mean_paragraph_len
↪  -is_quote , data = train)

summary(model4)

model4.1 <- lm(log(click_rate) ~ . -product -day_of_week -category  -mean_paragraph_len
↪  -is_quote - no_of_CTA , data = train)
model4.2 <- lm(log(click_rate) ~ . -product -day_of_week -category  -mean_paragraph_len
↪  -is_quote - is_image , data = train)
model5 <- lm(log(click_rate) ~ . -product -day_of_week -category  -mean_paragraph_len
↪  -is_quote - no_of_CTA - is_image , data = train)

anova(model4.1, model4)
anova(model4.2, model4)
anova(model5, model4)
```

```r
summary(model5)

par(mfrow=c(1,2))
plot(model5, which=1)
plot(model5, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   1286
```

```r
par(mfrow=c(1,2))
boxplot(model5$residuals, main="Residuals Boxplot")
plot(model5, which=4)
```

```r
par(mfrow=c(1,1))
```

Right Branch

```r
model6 <- lm(log(click_rate) ~
  sender + subject_len + body_len + is_weekend + times_of_day + no_of_CTA +
  mean_CTA_len + is_image + is_personalised + is_quote + is_emoticons +
  is_urgency + is_discount + is_price + target_audience +

  I(subject_len^2) + I(body_len^2) + I(no_of_CTA^2) + I(mean_CTA_len^2) +
  I(is_image^2) + I(is_quote^2) + I(is_emoticons^2) + I(is_price^2) +

  sender:subject_len + sender:body_len + sender:no_of_CTA + sender:mean_CTA_len +
↪  sender:is_emoticons + sender:is_price +
```

```
  subject_len:body_len + subject_len:is_weekend + subject_len:times_of_day +
  subject_len:no_of_CTA + subject_len:mean_CTA_len + subject_len:is_image +
  subject_len:is_quote + subject_len:is_emoticons + subject_len:is_urgency +
  subject_len:is_discount + subject_len:is_price + subject_len:target_audience +
  subject_len:is_personalised +

  body_len:is_weekend + body_len:times_of_day + body_len:no_of_CTA +
  body_len:mean_CTA_len + body_len:is_image + body_len:is_quote +
  body_len:is_emoticons + body_len:is_urgency + body_len:is_discount +
  body_len:is_price + body_len:target_audience + body_len:is_personalised +

  is_weekend:no_of_CTA + is_weekend:mean_CTA_len +
  is_weekend:is_image + is_weekend:is_quote + is_weekend:is_emoticons +
  is_weekend:is_price +

  times_of_day:no_of_CTA + times_of_day:mean_CTA_len + times_of_day:is_image +
  times_of_day:is_quote + times_of_day:is_emoticons + times_of_day:is_price +

  no_of_CTA:mean_CTA_len + no_of_CTA:is_image + no_of_CTA:is_quote +
  no_of_CTA:is_emoticons + no_of_CTA:is_urgency + no_of_CTA:is_discount +
  no_of_CTA:is_price + no_of_CTA:target_audience + no_of_CTA:is_personalised +

  mean_CTA_len:is_image + mean_CTA_len:is_quote + mean_CTA_len:is_emoticons +
  mean_CTA_len:is_urgency + mean_CTA_len:is_discount +mean_CTA_len:is_price +
  mean_CTA_len:target_audience + mean_CTA_len:is_personalised +

  is_image:is_personalised + is_image:is_quote + is_image:is_emoticons +
  is_image:is_urgency + is_image:is_discount + is_image:is_price +
  is_image:target_audience +

  is_personalised:is_quote + is_personalised:is_emoticons +
  is_personalised:is_price +

  is_quote:is_emoticons + is_quote:is_urgency + is_quote:is_discount +
  is_quote:is_price + is_quote:target_audience +

  is_emoticons:is_urgency + is_emoticons:is_discount + is_emoticons:is_price +
  is_emoticons:target_audience +

  is_urgency:is_price +

  is_discount:is_price +

  is_price:target_audience,

  data = train)
# In formal code, remove comments.
# stepAIC(model6, scope=list(upper=model6, lower = ~1), direction="both",
↪  k=log(nrow(train)), trace = FALSE)
# stepAIC(model6, scope=list(upper=model6, lower = ~1), direction="backward",
↪  k=log(nrow(train)), trace = FALSE)
```

```
model7 <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2), data = train)
summary(model7)
```

```
model8 <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2) + sender + is_weekend + is_personalised +
    is_price, data = train)
anova(model7,model8)
summary(model8)
```

```
par(mfrow=c(1,2))
plot(model7, which=1)
plot(model7, which=2)
```

```
par(mfrow=c(1,2))
boxplot(model7$residuals, main="Residuals Boxplot")
plot(model7, which=4)
```

```
par(mfrow=c(1,1))
```

```
par(mfrow=c(1,2))
plot(model8, which=1)
plot(model8, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   2, 1286
```

```
par(mfrow=c(1,2))
boxplot(model8$residuals, main="Residuals Boxplot")
plot(model8, which=4)
```

```
par(mfrow=c(1,1))
```

Validation

```
R2_list <- c()
Ra2_list <- c()
Cp_list <- c()
p_list <- c()
sigma2 <- sum(resid(model1)^2) / (nrow(train) - length(coef(model1)))
AIC_list <- c()
BIC_list <- c()
Pressp_list <- c()
MSE_list <- c()
valid1 <- valid[c(-168,-139,-220),] #new category and product in validation set

for (model in list(model3,model5,model7,model8)){
  model.summary <- summary(model)
  R2_list <- cbind(R2_list,model.summary$r.squared)
  Ra2_list <- cbind(Ra2_list,model.summary$adj.r.squared)
```

```
  Cp_list <-
↪  cbind(Cp_list,sum(resid(model)^2)/sigma2-(nrow(train)-2*length(coef(model))))
  p_list <- cbind(p_list,length(coef(model)))
  AIC_list <-
↪  cbind(AIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+2*length(coef(model)))
  BIC_list <-
↪  cbind(BIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+log(nrow(train))*length(coef(model))
  Pressp_list <- cbind(Pressp_list,sum((resid(model)/(1-influence(model)$hat))^2))
  MSE_list <- cbind(MSE_list,sum((valid1$click_rate -
↪  exp(predict(model,valid1)))^2)/(nrow(valid1)-length(coef(model))))
}
Cp_list <- rbind(Cp_list,p_list)
print(R2_list)
print(Ra2_list)
print(Cp_list)
print(AIC_list)
print(BIC_list)
print(Pressp_list)
print(MSE_list)
```

Outliers

```
hii <- function(model){
e<-model$residuals ##ordinary residuals
h<-influence(model)$hat ##diagonals of the hat matrix: a.k.a. leverage values
de<-e/(1-h) ##deleted residuals
plot(e,de, xlab="residuals", ylab="deleted residuals")
abline(0,1)
return(na.de=which(is.infinite(de)))
}
hii(model3)
```

```
hii(model5)
```

```
hii(model7)
```

```
hii(model8)
```

```
train_clean <- train[c(-2,-1286),]
model3_clean <- lm(log(click_rate) ~ . -product -day_of_week -category, data =
↪  train_clean)
model5_clean <- lm(log(click_rate) ~ . -product -day_of_week -category
↪  -mean_paragraph_len -is_quote - no_of_CTA - is_image , data = train_clean)
model7_clean <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2), data = train_clean)
model8_clean <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2) + sender + is_weekend + is_personalised +
    is_price, data = train_clean)
```

```
R2_list <- c()
Ra2_list <- c()
Cp_list <- c()
p_list <- c()
sigma2 <- sum(resid(lm(log(click_rate) ~ ., data = train_clean))^2) / (nrow(train) -
↪  length(coef(model1)))
AIC_list <- c()
BIC_list <- c()
Pressp_list <- c()
MSE_list <- c()
valid1 <- valid[c(-168,-139,-220),] #new category and product in validation set

for (model in list(model3_clean,model5_clean,model7_clean,model8_clean)){
  model.summary <- summary(model)
  R2_list <- cbind(R2_list,model.summary$r.squared)
  Ra2_list <- cbind(Ra2_list,model.summary$adj.r.squared)
  Cp_list <-
↪  cbind(Cp_list,sum(resid(model)^2)/sigma2-(nrow(train_clean)-2*length(coef(model))))
  p_list <- cbind(p_list,length(coef(model)))
  AIC_list <-
↪  cbind(AIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+2*length(coef(model)))
  BIC_list <-
↪  cbind(BIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+log(nrow(train))*length(coef(model))
  Pressp_list <- cbind(Pressp_list,sum((resid(model)/(1-influence(model)$hat))^2))
  MSE_list <- cbind(MSE_list,sum((valid1$click_rate -
↪  exp(predict(model,valid1)))^2)/(nrow(valid1)-length(coef(model))))
}
Cp_list <- rbind(Cp_list,p_list)
print(R2_list)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.3160159 0.3125819 0.3116118 0.3339379
```

```
print(Ra2_list)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2969634 0.2953964 0.2992683 0.3163368
```

```
print(Cp_list)
```

```
##         [,1]     [,2]     [,3]     [,4]
## [1,] 190.177 190.1356 172.3838 144.6417
## [2,]  41.000  37.0000  27.0000  39.0000
```

```
print(AIC_list)
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 409.3814 408.7883 390.8739 366.1113
```

```
print(BIC_list)
```

```
##          [,1]     [,2]     [,3]    [,4]
## [1,] 626.6453 604.8558 533.9502 572.777
```

```
print(Pressp_list)
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 1948.025 1945.212 1922.417 1890.217
```

```
print(MSE_list)
```

```
##            [,1]        [,2]        [,3]        [,4]
## [1,] 0.00619893 0.006134803 0.005577509 0.005878342
```

```
library(MASS)
for (model in list(model3_clean,model5_clean,model7_clean,model8_clean)){
  stu.res.del <- studres(model)
  print(head(sort(abs(stu.res.del), decreasing=TRUE)))
  }
```

```
##     1224      585     1767     1258     1766     1206
## 4.390268 3.271515 3.249164 2.924027 2.850823 2.805097
##     1224      585     1767     1258     1766     1206
## 4.325642 3.216214 3.138153 2.914086 2.851581 2.821976
##     1224      585     1816      766     1767     1766
## 4.217029 3.279481 2.931547 2.904914 2.890620 2.810420
##     1224      585     1767     1258      766     1816
## 4.460137 3.331927 2.959268 2.912703 2.851514 2.836196
```

```
qt(1-.5/(2*nrow(train_clean)), nrow(train_clean)-p_list-1) #Bonferroni's Threshold
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 3.592545 3.592521 3.592462 3.592533
```

```
for (model in list(model3_clean,model5_clean,model7_clean,model8_clean)){
  # h <- influence(model)$hat
  # p <- length(coef(model))
  # print(sort(h[which(h>2*p/n)], decreasing = TRUE))
  plot(model,which=4)
}
```

```
train_clean2 <- train_clean[-635,]

model3_clean2 <- lm(log(click_rate) ~ . -product -day_of_week -category, data =
↪  train_clean2)
model5_clean2 <- lm(log(click_rate) ~ . -product -day_of_week -category
↪  -mean_paragraph_len -is_quote - no_of_CTA - is_image , data = train_clean2)
model7_clean2 <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2), data = train_clean2)
model8_clean2 <- lm(formula = log(click_rate) ~ subject_len + body_len + times_of_day +
    no_of_CTA + is_emoticons + is_urgency + is_discount + target_audience +
    I(body_len^2) + I(no_of_CTA^2) + sender + is_weekend + is_personalised +
    is_price, data = train_clean2)

summary(model8_clean)
summary(model8_clean2)
```

```
R2_list <- c()
Ra2_list <- c()
Cp_list <- c()
p_list <- c()
```

```r
sigma2 <- sum(resid(lm(log(click_rate) ~ ., data = train_clean2))^2) /
↪  (nrow(train_clean2) - length(coef(model1)))
AIC_list <- c()
BIC_list <- c()
Pressp_list <- c()
MSE_list <- c()
valid1 <- valid[c(-168,-139,-220),] #new category and product in validation set

for (model in list(model3_clean2,model5_clean2,model7_clean2,model8_clean2)){
  model.summary <- summary(model)
  R2_list <- cbind(R2_list,model.summary$r.squared)
  Ra2_list <- cbind(Ra2_list,model.summary$adj.r.squared)
  Cp_list <-
↪  cbind(Cp_list,sum(resid(model)^2)/sigma2-(nrow(train_clean)-2*length(coef(model))))
  p_list <- cbind(p_list,length(coef(model)))
  AIC_list <-
↪  cbind(AIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+2*length(coef(model)))
  BIC_list <-
↪  cbind(BIC_list,nrow(train)*log(sum(resid(model)^2/nrow(train)))+log(nrow(train))*length(coef(model))
  Pressp_list <- cbind(Pressp_list,sum((resid(model)/(1-influence(model)$hat))^2))
  MSE_list <- cbind(MSE_list,sum((valid1$click_rate -
↪  exp(predict(model,valid1)))^2)/(nrow(valid1)-length(coef(model))))
}
Cp_list <- rbind(Cp_list,p_list)
print(R2_list)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.3160907 0.3126611 0.3116401 0.3339401
```

```r
print(Ra2_list)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2970271 0.2954657 0.2992886 0.3163268
```

```r
print(Cp_list)
```

```
##          [,1]     [,2]    [,3]     [,4]
## [1,] 186.9702 186.9034 169.265 141.6823
## [2,]  41.0000  37.0000  27.000  39.0000
```

```r
print(AIC_list)
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 409.1958 408.5941 390.7893 366.0827
```

```r
print(BIC_list)
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 626.4598 604.6616 533.8656 572.7484
```

```r
print(Pressp_list)
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 1947.827 1945.004 1922.322 1890.199
```

```r
print(MSE_list)
```

```
##              [,1]        [,2]        [,3]        [,4]
## [1,] 0.006198617 0.006134518 0.005577908 0.005878483
```

STA 206 Final Project Code - Visualizations

STA 206 | Fall 2024

Dec. 9, 2024

```
email <- read.csv('email/train_data.csv')
```

```
# Check for mmumber of missing values by column

missing_values <- colSums(is.na(email))
print(missing_values)
```

```
# Change categorical variables to factors
email$times_of_day <- as.factor(email$times_of_day)
email$sender <- as.factor(email$sender)
email$category <- as.factor(email$category)
email$product <- as.factor(email$product)
email$target_audience <- as.factor(email$target_audience)
email$day_of_week <- as.factor(email$day_of_week)
email$is_personalised <- as.factor(email$is_personalised)
email$is_discount <- as.factor(email$is_discount)
email$is_urgency <- as.factor(email$is_urgency)
email$is_weekend <- as.factor(email$is_weekend)

# number of unique values in each column
sapply(email, function(x) length(unique(x)))

# remove `is_timer` from the dataset
email <- email[, -which(names(email) == "is_timer")]

# Delete `campaign_id` column
email <- email[, -1]

# Subset based on `email$click_rate` not equal to 0
email_yes_click <- email[email$click_rate != 0, ]

# Keep full dataset for later use
email_full <- email
```

```
str(email)
```

```
# Illustrating issue with keeping `mail_full$click_rate` == 0

email_full_added_noise <- email_full
email_full_added_noise$click_rate <- email_full$click_rate + 0.0000000000001

model1_added_noise <- lm(click_rate ~ ., data = email_full_added_noise)
```

```r
par(mfrow=c(1,3))
plot(model1_added_noise, which=1)
plot(model1_added_noise, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   509, 697, 837, 876, 906, 907, 1109, 1162, 1387
```

```r
boxplot(model1_added_noise$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))

library(MASS)
boxcox(model1_added_noise)
```

```r
model2_added_noise <- lm(log(click_rate) ~ ., data = email_full_added_noise)
par(mfrow=c(1,3))
plot(model2_added_noise, which=1)
plot(model2_added_noise, which=2)
```

```
## Warning: not plotting observations with leverage one:
##   509, 697, 837, 876, 906, 907, 1109, 1162, 1387
```

```r
boxplot(model2_added_noise$residuals, main="Residuals Boxplot")
```

```r
par(mfrow=c(1,1))
```

```r
# Create a vector of the counts of zero and non-zero values of `click_rate`
click_rate_counts <- c(sum(email$click_rate == 0), sum(email$click_rate != 0))

# Create a vector of category names
categories <- c("Zero:", "Non-Zero:")

# Create a pie chart with percentages and labels
pie(click_rate_counts,
    labels = paste(categories, paste0(round(click_rate_counts/sum(click_rate_counts)*100,
    ↪  1), "%")),
    main = "Distribution of Click Rate Values",
    col = c("pink", "lightblue"))
```

```r
# Export as yes_no_click_piechart.png
```

```r
# Histogram of email_yes_click$click_rate

par(mfrow=c(1,2))

hist(email_yes_click$click_rate,
     main = "Histogram of Click Rate",
     xlab = "Click Rate",
     # col = "lightblue",
     border = "black")

hist(log(email_yes_click$click_rate),
     main = "Histogram of log(Click Rate)",
```

```
     xlab = "Log Click Rate",
     # col = "lightblue",
     border = "black")
```

```
par(mfrow=c(1,1))
```

```
str(email)
```

```
# Histogram of subject_len, body_len, mean_paragraph_len, no_of_CTA, mean_CTA_len,
↪  is_image, is_quote, is_emoticons, is_price

par(mfrow = c(3, 3))  # Set up a 3x3 grid for the plots

# Subject Length
hist(email_yes_click$subject_len,
     main = "Subject Length",
     xlab = "Subject Length",
     # col = "lightblue",
     border = "black")

# Body Length
hist(email_yes_click$body_len,
     main = "Body Length",
     xlab = "Body Length",
     # col = "lightblue",
     border = "black")

# Mean Paragraph Length
hist(email_yes_click$mean_paragraph_len,
     main = "Mean Paragraph Length",
     xlab = "Mean Paragraph Length",
     # col = "lightblue",
     border = "black")

# Number of CTAs
hist(email_yes_click$no_of_CTA,
     main = "Number of CTAs",
     xlab = "Number of CTAs",
     # col = "lightgreen",
     border = "black")

# Mean CTA Length
hist(email_yes_click$mean_CTA_len,
     main = "Mean CTA Length",
     xlab = "Mean CTA Length",
     # col = "lightgreen",
     border = "black")

# Is Image
hist(email_yes_click$is_image,
     main = "Is Image",
     xlab = "Is Image",
     # col = "lightcoral",
```

```r
     border = "black")

# Is Quote
hist(email_yes_click$is_quote,
     main = "Is Quote",
     xlab = "Is Quote",
     # col = "lightcoral",
     border = "black")

# Is Emoticons
hist(email_yes_click$is_emoticons,
     main = "Is Emoticons",
     xlab = "Is Emoticons",
     # col = "lightpink",
     border = "black")

# Is Price
hist(email_yes_click$is_price,
     main = "Is Price",
     xlab = "Is Price",
     # col = "lightpink",
     border = "black")
```

```r
panel.cor <- function(x, y){
  #usr <- par("usr")
  #on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y, use="complete.obs"), 2)
  txt <- paste0("R = ", r)
  cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(~ log(click_rate) + subject_len + body_len + mean_paragraph_len + no_of_CTA +
↪  mean_CTA_len + is_image + is_quote + is_emoticons + is_price,
      data = email_yes_click,
      lower.panel = panel.cor,
      main = "Scatterplot Matrix of Quantitative Variables")
```

```r
# Boxplot for click_rate based on sender
boxplot(log(click_rate) ~ sender,
        data = email_yes_click,
        main = "Log Click Rate by Sender",
        xlab = "Target Sender",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$sender))),
        names = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11"))
```

```r
# Boxplot for click_rate based on day_of_week
boxplot(log(click_rate) ~ day_of_week,
        data = email_yes_click,
        main = "Log Click Rate by Day of Week",
        xlab = "Day of Week",
```

```
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$sender))),
        names = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
        ↪  "Sunday"))
```

```
# Boxplot for click_rate based on is_weekend
boxplot(log(click_rate) ~ is_weekend,
        data = email_yes_click,
        main = "Log Click Rate by Weekend Indicator",
        xlab = "Is Weekend",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_weekend))),
        names = c("Weekday (0)", "Weekend (1)"))
```

```
# Boxplot for click_rate based on times_of_day
boxplot(log(click_rate) ~ times_of_day,
        data = email_yes_click,
        main = "Log Click Rate by Time of Day",
        xlab = "Time of Day",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$times_of_day))),
        names = c("Morning", "Noon", "Evening"))
```

```
par(mfrow=c(1,3))

boxplot(log(click_rate) ~ day_of_week,
        data = email_yes_click,
        main = "Log Click Rate by Day of Week",
        xlab = "Day of Week",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$sender))),
        names = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
        ↪  "Sunday"))

# Boxplot for click_rate based on times_of_day
boxplot(log(click_rate) ~ times_of_day,
        data = email_yes_click,
        main = "Log Click Rate by Time of Day",
        xlab = "Time of Day",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$times_of_day))),
        names = c("Morning", "Noon", "Evening"))

# Boxplot for click_rate based on is_personalised
boxplot(log(click_rate) ~ is_personalised,
        data = email_yes_click,
        main = "Log Click Rate by Personalised (Y/N)",
        xlab = "Personalised (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_personalised))),
        names = c("Not Personalised (0)", "Personalised (1)"))
```

```r
# Boxplot for click_rate based on category
boxplot(log(click_rate) ~ category,
        data = email_yes_click,
        main = "Log Click Rate by Category",
        xlab = "Category",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$category))),
        names = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
        ↪  "13", "14", "15"))
```

```r
# Boxplot for click_rate based on is_personalised
boxplot(log(click_rate) ~ is_personalised,
        data = email_yes_click,
        main = "Log Click Rate by Personalised (Y/N)",
        xlab = "Personalised (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_personalised))),
        names = c("Not Personalised (0)", "Personalised (1)"))
```

```r
# Boxplot for click_rate based on is_discount
boxplot(log(click_rate) ~ is_discount,
        data = email_yes_click,
        main = "Log Click Rate by Discount Indicator",
        xlab = "Discount (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_discount))),
        names = c("No Discount (0)", "Discount (1)"))
```

```r
# Boxplot for click_rate based on is_urgency
boxplot(log(click_rate) ~ is_urgency,
        data = email_yes_click,
        main = "Log Click Rate by Urgency Indicator",
        xlab = "Urgency (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_urgency))),
        names = c("0", "1"))
```

```r
# Boxplot for click_rate based on target_audience
boxplot(log(click_rate) ~ target_audience,
        data = email_yes_click,
        main = "Log Click Rate by Target Audience",
        xlab = "Target Audience",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$target_audience))),
        names = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
        ↪  "13", "14", "15", "16"))
```

```r
par(mfrow=c(1,3))

# Boxplot for click_rate based on is_weekend
boxplot(log(click_rate) ~ is_weekend,
        data = email_yes_click,
        main = "Log Click Rate by Weekend Indicator",
```

```
        xlab = "Is Weekend",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_weekend))),
        names = c("Weekday (0)", "Weekend (1)"))

# Boxplot for click_rate based on is_discount
boxplot(log(click_rate) ~ is_discount,
        data = email_yes_click,
        main = "Log Click Rate by Discount Indicator",
        xlab = "Discount (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_discount))),
        names = c("No Discount (0)", "Discount (1)"))

# Boxplot for click_rate based on is_urgency
boxplot(log(click_rate) ~ is_urgency,
        data = email_yes_click,
        main = "Log Click Rate by Urgency Indicator",
        xlab = "Urgency (Y/N)",
        ylab = "Log Click Rate",
        col = rainbow(length(levels(email_yes_click$is_urgency))),
        names = c("0", "1"))
```

‘‘‘