Will a User Click on a Marketing Email?

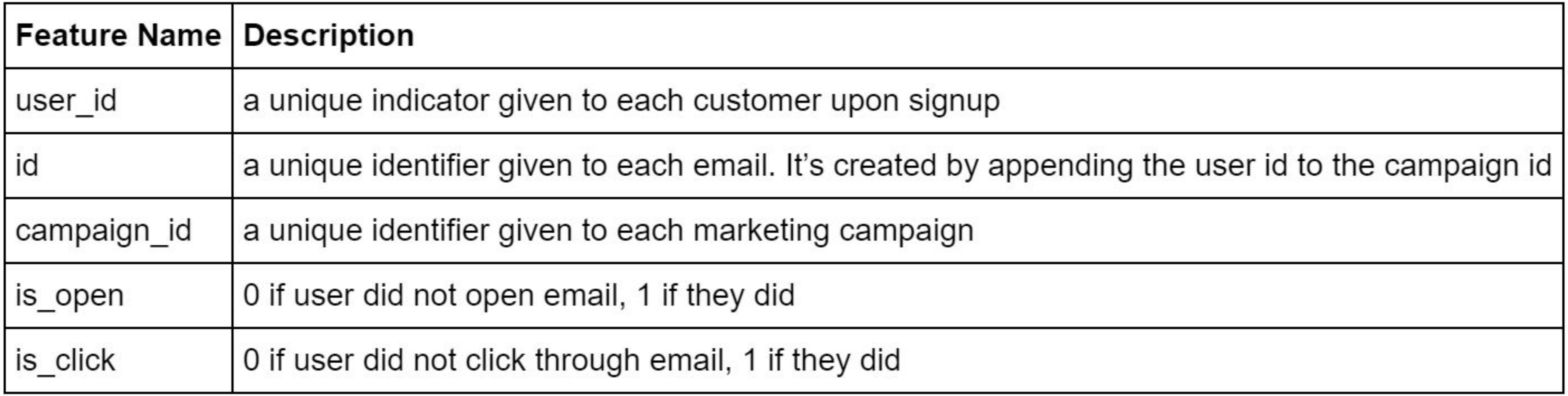
The Problem

Email Marketing is still the most successful marketing channel and the essential element of any digital marketing strategy. Marketers spend a lot of time in writing that perfect email, laboring over each word, designing catchy layouts on multiple devices to get them higher levels of open and click rates. In this notebook, I’ll analyze user data and marketing data to attempt to predict whether a user will click through any given email.

1) User level data - collected by Analytics Vidhya on their subscribers

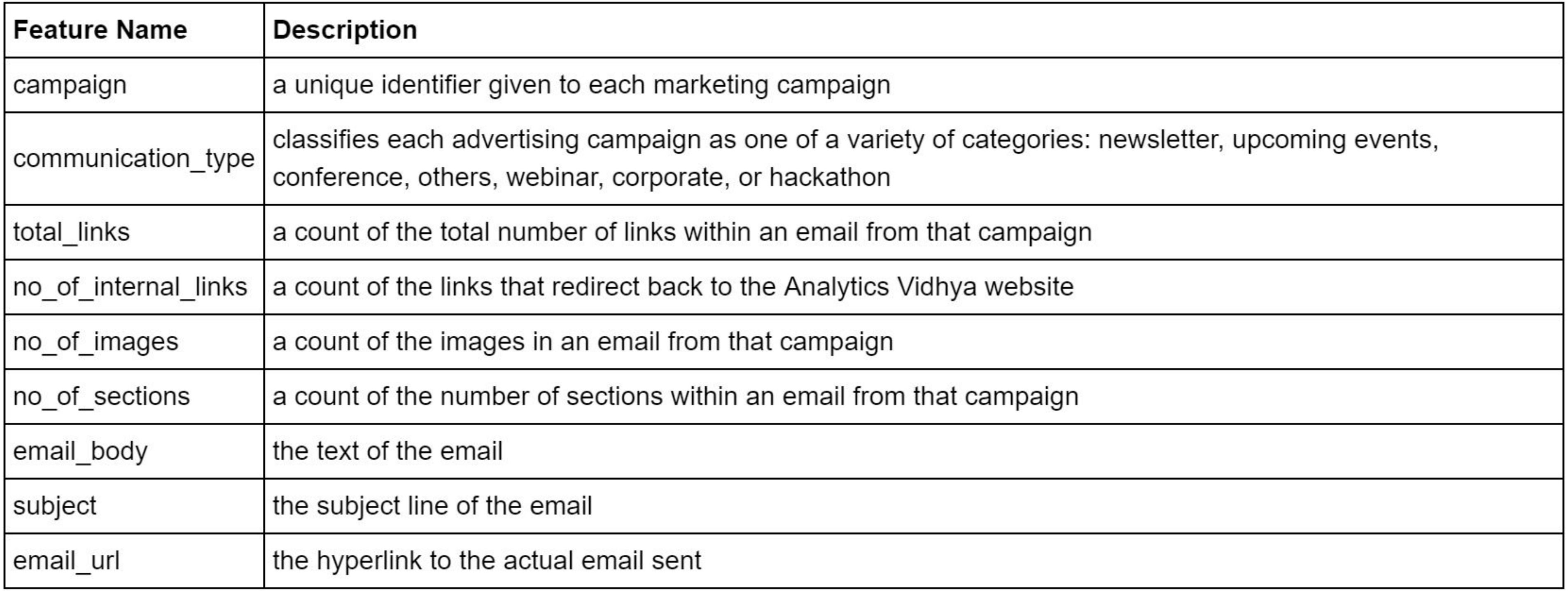
2) Campaign level data - contains information about the marketing campaign and the email that was sent to the users

User Features

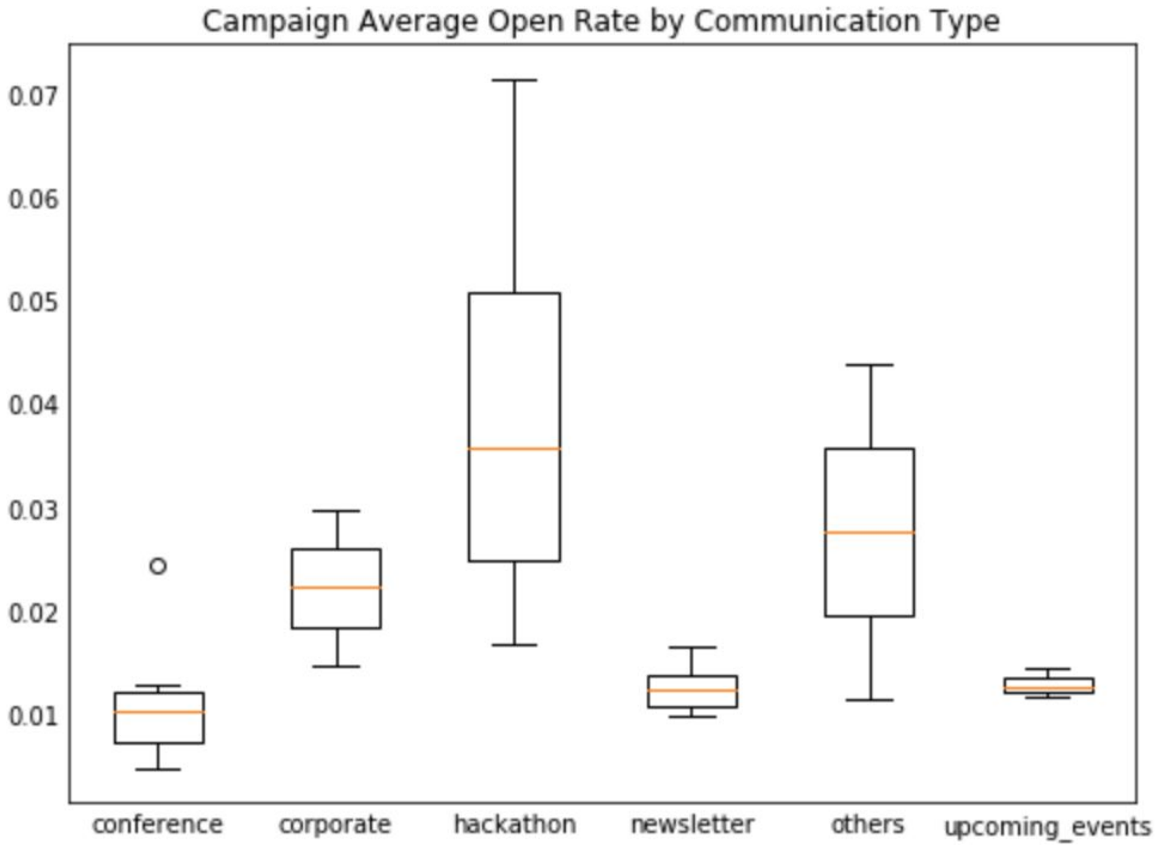


Feature Name	Description
user_id	a unique indicator given to each customer upon signup
id	a unique identifier given to each email. It’s created by appending the user id to the campaign id
campaign_id	a unique identifier given to each marketing campaign
is_open	0 if user did not open email, 1 if they did
is_click	0 if user did not click through email, 1 if they did

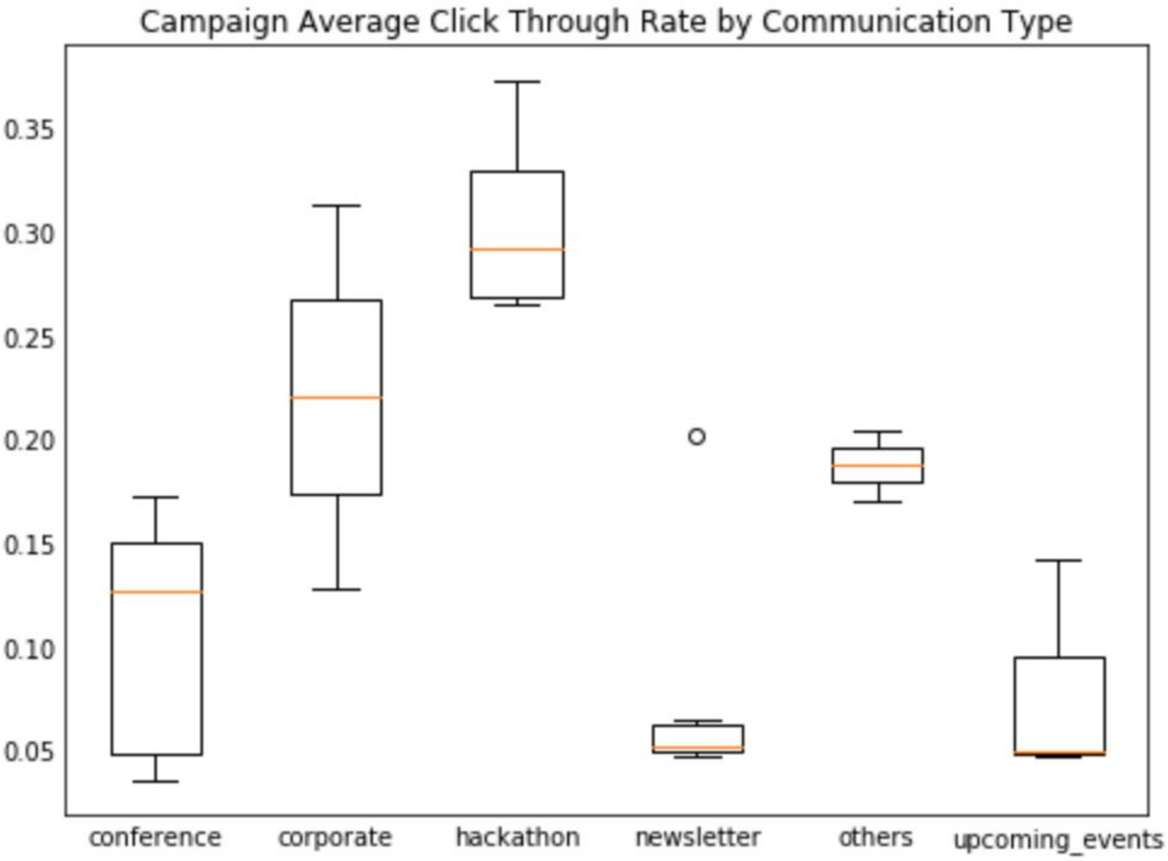
Campaign Features



Feature Name	Description
campaign	a unique identifier given to each marketing campaign
communication_type	classifies each advertising campaign as one of a variety of categories: newsletter, upcoming events, conference, others, webinar, corporate, or hackathon
total_links	a count of the total number of links within an email from that campaign
no_of_internal_links	a count of the links that redirect back to the Analytics Vidhya website
no_of_images	a count of the images in an email from that campaign
no_of_sections	a count of the number of sections within an email from that campaign
email_body	the text of the email
subject	the subject line of the email
email_url	the hyperlink to the actual email sent



The emails that get opened most are in the categories hackathon, corporate, and others.



The emails that get clicked the most are from the categories hackathon, corporate, others and conference.

Feature Engineering

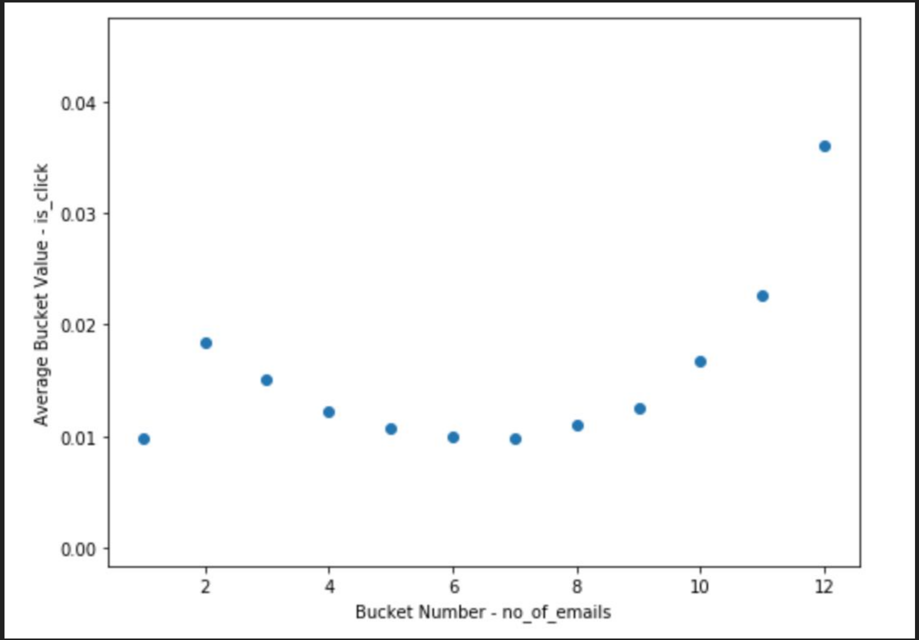
- Day of week
- Hour of day
- Percentage of emails received that are of each communication type
- Cumulative count of emails received
- Total count of emails received
- NLP

NLP and its Shortcomings

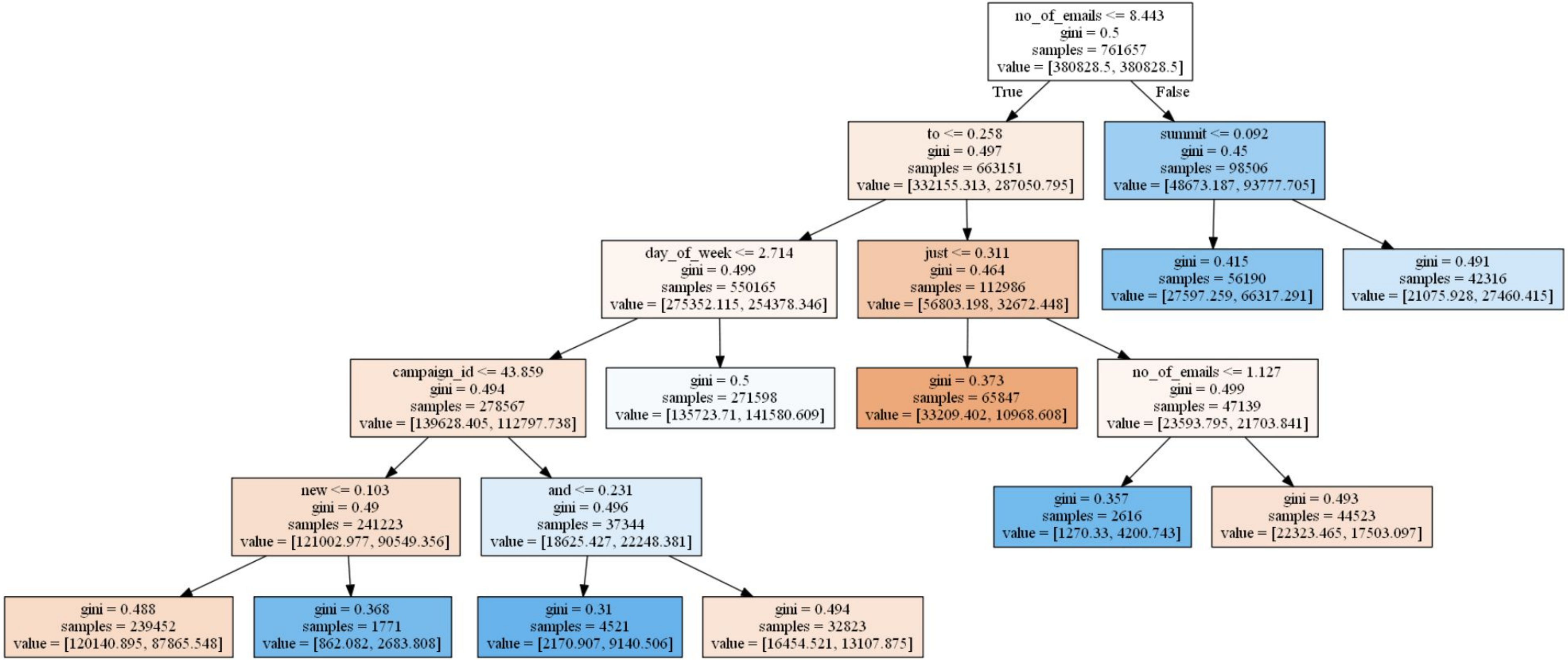
I performed NLP on the email subject lines to determine which words are indicative of clicks. There were serious limitations in the way I created my train and test sets. I used a 75/25 split based on time, which had the consequence of leaving only three campaigns in my test set. That being, there were only three subject lines on which to perform NLP in the test set. The result was a small subset of words, and poor indicators. In the future, I would split on a different criteria, or wait for more data.

Decision Tree

I used a decision tree algorithm to determine the most useful features. The total number of emails was the first split. Those having received more emails tended to click more frequently. Other important features included the features generated by NLP.



Decision Tree



no_of_emails <= 8.443
gini = 0.5
samples = 761657
value = [380828.5, 380828.5]

True

False

to <= 0.258
gini = 0.497
samples = 663151
value = [332155.313, 287050.795]

summit <= 0.092
gini = 0.45
samples = 98506
value = [48673.187, 93777.705]

day_of_week <= 2.714
gini = 0.499
samples = 550165
value = [275352.115, 254378.346]

just <= 0.311
gini = 0.464
samples = 112986
value = [56803.198, 32672.448]

gini = 0.415
samples = 56190
value = [27597.259, 66317.291]

gini = 0.491
samples = 42316
value = [21075.928, 27460.415]

campaign_id <= 43.859
gini = 0.494
samples = 278567
value = [139628.405, 112797.738]

gini = 0.5
samples = 271598
value = [135723.71, 141580.609]

gini = 0.373
samples = 65847
value = [33209.402, 10968.608]

no_of_emails <= 1.127
gini = 0.499
samples = 47139
value = [23593.795, 21703.841]

new <= 0.103
gini = 0.49
samples = 241223
value = [121002.977, 90549.356]

and <= 0.231
gini = 0.496
samples = 37344
value = [18625.427, 22248.381]

gini = 0.357
samples = 2616
value = [1270.33, 4200.743]

gini = 0.493
samples = 44523
value = [22323.465, 17503.097]

gini = 0.488
samples = 239452
value = [120140.895, 87865.548]

gini = 0.368
samples = 1771
value = [862.082, 2683.808]

gini = 0.31
samples = 4521
value = [2170.907, 9140.506]

gini = 0.494
samples = 32823
value = [16454.521, 13107.875]

Logistic Regression

Because the decision tree wasn’t very successful at predicting whether a user would click, I opted for a logistic regression model, which predicted with about 40% accuracy.

Conclusions

The accuracy obtained by my models is not good. I would suggest further exploration, more data, and tackling it again with a team with more knowledge of complex modeling techniques. The challenges I faced were largely due to my inexperience with deriving insights from simple datasets.