loxz digital

Sentiment Analysis Model For Email

By Jeffery Ott
Data Scientist

Table of Contents

INTRODUCTION

Often it can be challenging to find the correct phrasing to increase customer engagement. As part of our Loxz Portfolio, we introduce you to v1.1 of the Sentiment Analysis predictive analytics model for Email. The Sentiment Analysis RealTime Model aims to help alleviate the challenge of finding the optimal tone for the target metric you are looking to maximize.

The model is a two-part model, meaning during the first step, the text from email is parsed and loaded into our sentiment model. Part one is a BERT model, which analyzes the text in a bidirectional format and returns a multi-label classification of 8 different tones. These tones will evolve over time and subsequent builds of the model can be tested with different tones based on client requirements. The second step is utilizing the 8-tone classification in a metric-specific random forest. The output of this random forest is the desired target variable and confidence of that metric or target variable and how to modify the model through various inputs to maximize the target metric. The architecture is in place and can be trained to whichever email corpus and target metric the campaign manager chooses.

DATASETS FOR SENTIMENT ANALYSIS MODEL

The training dataset is also split into two parts. The first data set is the data on which the BERT model was trained. This data set is comprised of 2762 text columns, each with one of the following emotional labels: See Fig. 1

The second data set is the proprietary email data with the generated target variables. There are 4128 emails, each belonging to a specific industry and campaign type. For each of these emails, industry benchmarks have been assumed for each of our target variables have been taken from the following site https://mailchimp.com/resources/email-marketing-benchmarks. The targets were then generated with a normal distribution around their means with the appropriate variances. The targets and their ranges are as follows:

- **Click To Open Rate (CTOR) 6% -17% (varied by industry) (st_dev 2%)**
- **Revenue Generate Per Email 2-14 (st_dev 2$))**
- **Conversion Rate 2%-5% (st_dev 1%)**

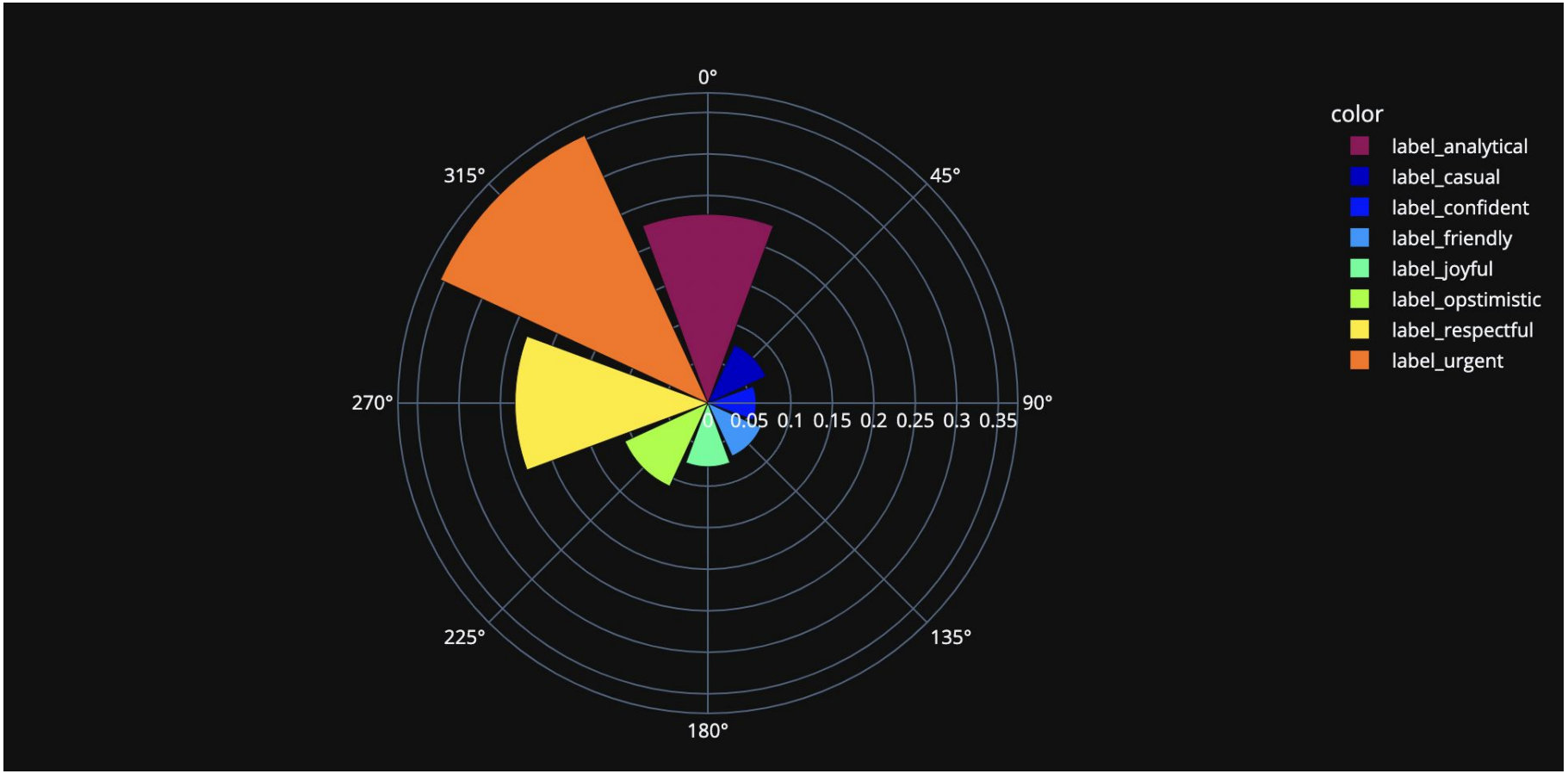

Figure 1. Example emotional sentiment from email

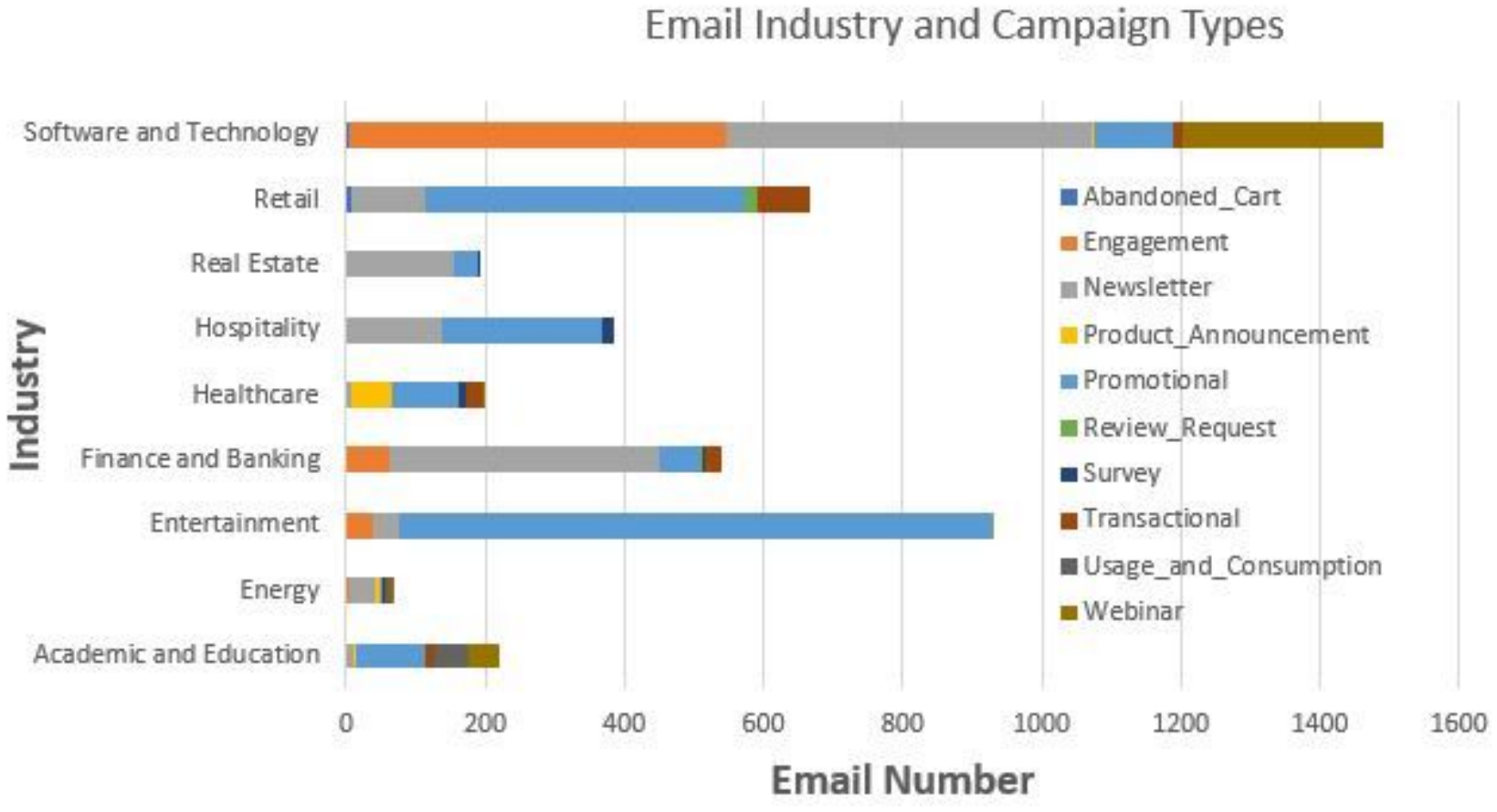

Figure 2. Email Dataset Values

DATA PREPARATION, ENGINEERING AND WANGLING

The data inputs consist of three primary sources from which assumptions are drawn. The first is the MailChimp data. This source provides Industry differentiated target metrics from which average CTR and average open rate are provided. From these variables, the average click-to-open rate was found. For revenue per email the following site was referenced https://ecommercefastlane.com/how-to-calculate-your-brands-revenue-per-email-send. Finally for the final metric of conversion rate the following website was referenced https://marketinginsidergroup.com/content-marketing/email-marketing-conversion-rate-comparison/

The targets were then generated from a normal distribution with variances discussed in the feature engineering section. The tone data is a dataset comprised of 2762 phrases with their corresponding labels. To prepare the data, the labels were one-hot encoded to their designated label, and the text was run through a Bert text encoder which outputs the corresponding tokens and attention mask.

Finally, the email data set is a corpus of 4128 emails whose Industry and campaign types have been labeled. A summary of all the data is given in the following table.

TABLE I: Data-set meta data and the corresponding value for campaign type and industry type.

Metadata	Meaning
Tones	[label_analytical,label_casual,label_confide label_friendly,label_joyful,label_optimistic, label_respectful, label_urgent]
Revenue_Per_Email	Continuous value for a revenue generated event
Click_to_Open_Rate	Continuous value for a click-through event
Conversion_Rate	Continuous value for a conversion event
campaign types	[Abandoned Cart, Newsletter, Promotional, Survey, Transactional, Webinar, Engagement, Review_Request, Product Announcement]
industry types	Software and Technology, Academic and Education, Entertainment, Finance and Banking, Hospitality, Real Estate, Retail, Energy, Healthcare

FEATURE ENGINEERING

To simulate real-world campaigns, our synthetic variables, the target features, were randomly assigned a continuous variable based on the above mentioned parameters. In order to find the target variable **click to open rate** the following formula is applied:

(Click Rate)/(Open Rate)=CTOR

This metric is run through a randomized normal distribution based on industry averages for each email type in a specific Industry. For the other target metrics, a normal distribution is simulated from which the variables are generated. The distribution uses the middle of the given range as the mean and half of the difference between the mean and limits is the standard deviation.

MODEL ALGORITHMS

As the data set is quite small, a Pre-trained Bert model was used to encode and process the data. This dataset was passed into a PyTorch lightning network, and the labels were predicted. The model's tone predictions were then fed into a random forest model that also included the industry and Campaign type. Since a mixture of continuous and binary variables was present, a random forest was a good starting point. Future work will be to implement different algorithms such as boosting or Neural networks to improve target prediction variances. The goal is to get a large enough variance between each of the recommendations in the output. In a subsequent version of this model, we will introduce auto-generation of language to match each tone. Sparse resources are available for tone generation currently. But' its good to stay ahead of the curve. An automated response, based on tonality that the campaign engineer desires. This has been discussed with the data science team and we are working on utilizing and finding tools to auto-generate text that increases the desired target variable upon "run."

Model Algorithms used in this Model:

- Bert_Uncased_L 2_H 128_A-2
- NN with a single Linear Layer
- Random Forest

MODEL DEVELOPMENT

The first step of Model Development is the training of the Bert model. The text of each email was encoded with the Bert encoder. Due to the length of the training phrases, the encoder only encoded 100 words at a time as the "shape" and along with a given attention mask. The encoded data was then fed into a pre-trained bert_uncased_L 2_H 128_A-2 model to output to a linear layer which brought the bert output to the target space with eight different label probabilities.

These outputs were then run through a Sigmoid and converted to probability space. Binary cross entropy loss was used in predicting each of the labels, and a label probability vector with length 8 was the output. This model was trained using PyTorch lightning to distribute the training and monitor the loss. The learning rate was also scheduled to help pinpoint the minimum loss. Once the minimum loss was found, the weights and biases were saved. The model achieved a label prediction accuracy of 89%

The email data was then loaded, and parsed using Beautiful Soup and Regex. From this pre-processing, the body of the email was extracted. The body was then fed into the frozen model, and then each sentence is tokenized and the sentiment is found, being aggregated across the email. This allows the classification of each tone for the 4128 emails. The label and campaign type were one-hot encoded, and the algorithm was then run through a random forest, each of which were trained on a separate target variable. From this group of random forests, the correct model is selected to give the desired target output. The different pretrained models are listed in Table II.

TABLE II: Different trained Random Forest models

Target	Model Name
Open_Rate	ORM
Click_Through_Rate	CRM
Unsubscribe_Rate	URM
Bounce_Rate	BRM
Click_To_Open_Rate	CTORM
Conversion_Rate	ConM
Revenue_Per_Email	RVM

DEVELOPMENT ENVIRONMENT

To further prepare our model for development, we utilized AWS Sagemaker for training and deployment. The training and hyper-parameter optimization procedure will be conducted in AWS Sagemaker using Jupyter Sagemaker Notebook Instances. The training data will be hosted on S3 Bucket which keep tracks of the training jobs and as well log files. Sagemaker is also recognized as our deployment platform.

DEVELOPMENT ASSUMPTIONS AND VALIDATION

The model is built in Amazon SageMaker. The assumptions are including and not limited to the following:

- The tones are treated as a ground truth this could contribute to some mis-classification of the tone
- The target variables were generated and the model would need to be retrained and reexamined with the introduction of client campaign data
- There is an assumption that the data set is the correct size necessary for these types of classification tasks.
- There is an assumption that the best email performance is in part based on the tone, and the recommendations are an attempt to align with the tone of the highest performing email for the given target metric and email type.

This is model version 1.1. It has been a collaborative effort over the last 6 months beginning as a sentiment analysis engine and turning into a recommendation engine with 3 to 5 outputs.

MODEL METRICS

There are several different outputs of the model and each can be customized to the desired endpoint/client. Keep in mind, we only serve predictions, and do not make recommendations on how the output should be displayed a clients UI.

The first output is the 95% confidence interval of the Random Forest prediction. [Fig.3] This is done using previously calculated error quartiles. From the available data, there is a belief that the true prediction in the given interval is true 95% of the time.

We serve three predictions upon the declaration of a target variable after the model has been run, during her workflow.

The next output is the recommendations. Fig.4 These are based on the optimal case for your campaign, industry and target. The green is your current tone and the red is the recommendation in order to get your tone closer to the optimal tone for that classification. If there is no available data on the optimal tone for the target industry and campaign type this data is not displayed.

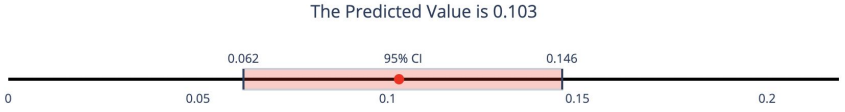

Figure 3. The predicted value of click-to-open-rate

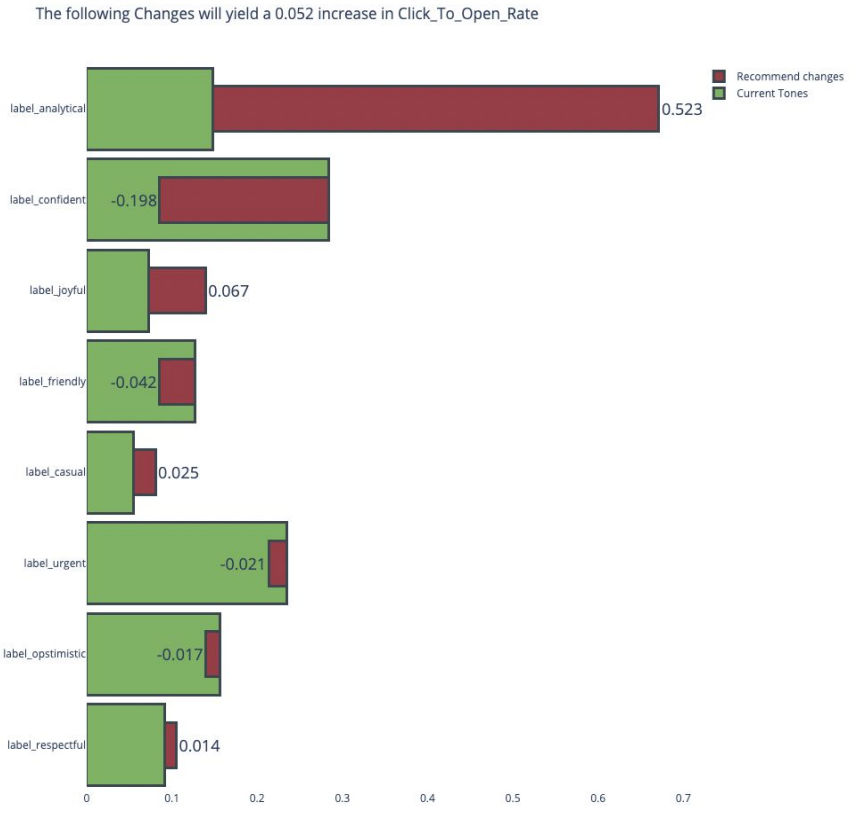

Figure 3. The predicted value of click-to-open-rate

Finally the text output at the bottom is the cleaned email whose shade is indicative of how far from the actual tone. In order to clarify ambiguity the actual distance from the optimal tone is also printed at the end of each sentence. This gives the campaign manager different passages/options to focus on when making corrections to the email text editor. The outputs can all be aggregated and explaining the model this way does provide the engineer confidence on how to proceed with message tonality.

DATA SCIENTIST OWN CONCLUSIONS

This model serves as a proxy for future email campaigns. There are some large assumptions made, and the results are difficult to prove as the mean is usually predicted and the standard deviation is the error. That being said, the architecture is in place, and with the addition of more data, this model is capable of potentially helping email campaign engineers to find the right way to phrase their email.

```
 building solutions that work smarter and faster for your business if you can dream it  we can build it .|70|  life
with an agile team ever consider working with an innovative agile team  what to her first hand what that is like pr
oduct manage  ally kumar shares her experience working with an agile and innovative team  when youve worked with te
ams who are attempting to implement an agile methodology without making significant efforts to adjust waterfall cul
ture you can expect to run into all sorts of growing pains .|41|  business requirements that keep pace with custome
rs but slow down with it implementation being the most obvious .|58|  these same experiences can cause you to be un
necessarily suspect when things seem to be running quietly on their own with truly innovative  agile team .|54|   t
o see more of what ally has to say visit here .|71|   for more information facebook twitter linkedin youtube instag
ram email .|66|
```

CONCLUSION AND FUTURE CONSIDERATIONS

We constructed a model for a synthetic email campaign dataset for real-time optimizations. These predictions are served to endpoints in milli-seconds. The data is a proxy for real data, which the model pipeline is ready to train and accept.

The next version of this model could utilize additional retrained sentiment scores to understand the validity of the sentiment model better. Opportunities to use ML to optimize text upon run, based on inputs is our desired next step. The model upon run, will automatically create the proper tone needed in a grammatically correct manner to generate the desired sentiment. The option is also available for merging two or more models. When you merge multiple models together to optimize text or images further, you will likely achieve enhanced metrics for your target variable. We call this a multi-modal build. For example, if we were to serve character count predictions coupled with sentiment analysis predictions we would likely achieve a higher target variable metric. You can read more about all of our models here: https://loxz.com/#/model-portfolio.

REFERENCES

https://stats.stackexchange.com/questions/56895/do-the-predictions-of-a-random-forest-model-have-a-prediction-interval

https://andrewpwheeler.com/2022/02/04/prediction-intervals-for-random-forests/

https://ecommercefastlane.com/how-to-calculate-your-brands-revenue-per-email-send/

https://mailchimp.com/resources/email-marketing-benchmarks