Discount Optimization Model For Email

By Buwani Manuweera
Data Scientist

loxz digital

Table of Contents

ABSTRACT

This report describes the Discount Optimization model developed by Loxz Digital. This model provides predictive analytics of product discounts prior to deploying a campaign so that the possible outcomes can be improved based on the model's predictions and recommendations. The model provides a discount recommendation with the highest email engagement rates based on the selected target variable, preferred discount range, industry, and campaign type.

The dataset contains 1183 samples of data. The algorithm used in this model is XGBoost regression, which is an ensemble of Decision Trees. The current model is able to provide the highest accuracy of 92.8% with room for further improvement.

I . INTRODUCTION

The Discount Optimization model is developed to provide predictive analytics on product discounts offered in email campaigns. The model provides predictions on an email campaign for the selected esmail engagement metrics as well as recommendations on how to optimize the discounts in emails in order to maximize the specified targeted campaign metrics. These discount emails can belong to any campaign type in any industry the campaign engineer uses as inputs.

Using our Discount Optimization model, the users can identify the optimum ranges of discounts to be used in the campaign in real-time to achieve the best possible outcome.

Ⅱ. DATA SETS USED

The model uses 1183 emails across nine industries and nine campaign types. Table 1 shows the distribution of emails across industries and Figure 1 shows the percentages of emails across the industries considered for the model. A collection of emails transcend different steps of filtering to select the suitable emails for the discount optimization model. The filtration process of emails first filters out emails with only images, and then emails without any discount offers in % values. This results in the set of emails with discounts offered as a percentage for the current model.

The types of campaigns are as follows: Currently, there is a higher number of emails for promotional campaigns as the discount offers are usually for promotions. But we plan to include more emails for other campaigns as we gather more data.

- Abandoned Cart
- Newsletter
- Promotional
- Survey
- Transactional
- Event Promotion
- Transactional
- Webinar
- Engagement
- Review Request
- Product Announcement

The dataset uses the plain text from each email and parses it to get the features for the model. The features include the *discount amount, industry type, campaign type, number of the discount mentions in the email body, and word embeddings*.

The target variables considered are *Open Rate, Click-To-Open Rate, Conversion Rate, and Revenue per email*. The current model is designed for *Open Rate, Click-To-Open Rate, and Conversion Rate* and *Revenue per email* will be implemented as the next step.

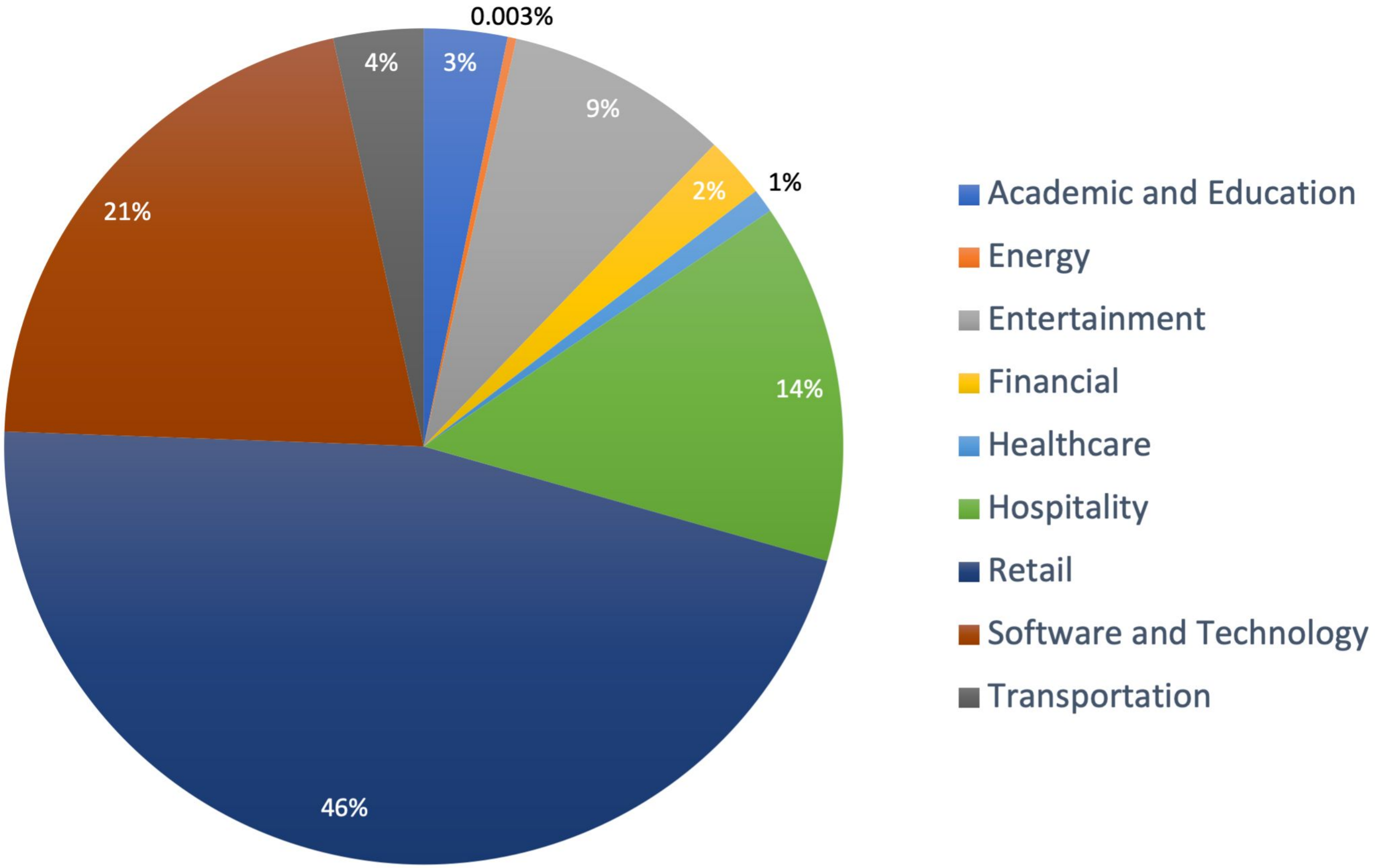


Figure 1. Percentages of emails distributed across all industries for the model

Industry	No. of Emails
Academic and Education	38
Energy	4
Entertainment	102
Financial	28
Healthcare	11
Hospitality	165
Retail	546
Software and Technology	248
Transportation	41
Total	1183

Table 1. Email distribution in the dataset across the industries (after resample)

III. . DATA SOURCES

Due to the lack of publicly available data for the target variables, we refer to multiple resources of email campaign benchmarks to generate a set of data within predefined normalized distribution ranges [1–3, 5, 8]. This approach allows us to create our model for potential use in actual email campaigns. Currently we use resources from Campaign Monitor, Ruler Analytics, ContentGrip, Listrak's Cross-Channel Marketing Automation Platform and CM Commerce.

For the email data samples, a collection of carefully curated emails belonging to different campaigns and industries is used in the '.eml' format. That will provide the plain text of each email to be considered for feature engineering.

IV. TOOLS REQUIRED

The implementation of the model was done in the Jupyter Notebook instance in *AWS SageMaker* using Python programming [6]. The machine learning model development was done entirely using the Scikit-learn package for Python [9]. The word embeddings used as part of the features for the model were collected using the *Sentence-BERT* tool developed by Reimers et al. [10]. Other tools were used for parsing purposes such as Beautiful Soup.

V. FUTURE ENGINEERING

As part of the synthetic features used, *campaign type* and industry type are used directly from the dataset as explained in Section 2. The emails in the dataset belong to these industries and campaigns and using them as features provide that information into the model.

Apart from them, the *discount value* given in the email is also extracted. This is an important feature for the Discount Optimization model as the model predictions and recommendations rely on the discount value in the email. To extract the discount value from each email, the text is first scanned carefully using regular expressions. In this process, it looks for the percentage sign ('%') within the text and checks if there are any digits preceding. This filters out the discounts offered in '$' amounts as the current model focuses on percentage discounts.

Not all percentage values indicate a discount offer in the email and therefore, more filters are used to get the values specifically given with discount offers. This is done by searching the text for specific words/phrases given below.

- % off
- Save % on
- Discount

Only the percentage values mentioned with the above words/phrases were considered for the discount feature. This filtering process was especially challenging given the HTML tags in the email text occurring in between discount values and, getting the correct discount value by ignoring other percentages mentioned in the text. Therefore, a proper text filtering process and regular expressions were important for accurate discount extractions. In case there is more than one discount offer in one email, the maximum discount value is considered for the feature in the current model.

The model also uses the frequency of the discount value mentioned in the email body as that could influence the Click-to-open rate and conversion rate. This is used to give additional context to help improve prediction performance.

The remaining features are the word embeddings for the email body. For that, *Sentence-BERT* was used to get the embeddings for the entire email as a feature vector [10]. *Sentence-BERT* provides a feature vector of 384 values for each email representing the information in the entire email.

VI. MODEL DEVELOPMENT

As the model predicts continuous values, we considered regression algorithms for machine learning. The regression model takes the set of features described in Section 5 along with the target variables explained in Section 2 as inputs and trains the model for predictions. The model takes a total of 388 features and three different target variables. The model is trained on one selected target variable at a time based on the user's inputs.

Multiple regression algorithms were considered for experimenting with the model accuracy and based on the results, *XGBoost* was selected due to its higher overall performance [4]. All the algorithms were used with their default hyperparameter values for uniform analysis. The results are further explained in Section 9. The model is trained using 90% of the data while the performance is evaluated using the remaining 10%.

The discount model is developed to take the user inputs and output the model accuracy, prediction for the user email, and recommendations. The model is trained for the target variable the user has selected and provides the prediction for the user's email. Based on the model prediction, the three best recommendations for discount values within the user's desired discount range are selected for output.

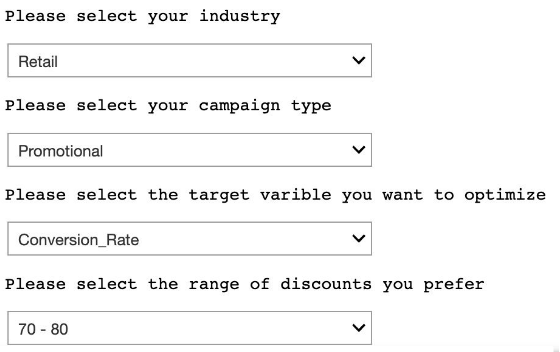
VII. ALGORITHMS USED

For getting embeddings, the model uses Sentence-BERT [10]. It is developed to get embedding for sentences/paragraphs which are used here to get the embeddings of the entire email. This provides a vector representation of the text document with its semantic relations in the sentences yielding more information on the meaning of the text.

For machine learning, the model uses XGBoost algorithm [4]. The tree-based algorithms are easier to interpret than other algorithms. XGboost is a tree-based ensemble method similar to Random Forest [7]. Random forest uses a bagging method where the model output is based on the majority prediction of the trees. XGBoost is a boosting method where the trees are built in series and the current tree's results are used to build the next tree to minimize the current loss. This will eventually give an improved tree with better performance using the results of a series of trees.

VIII. USE CASES

The discount optimization model is developed for email marketing campaigns and is to be used by the campaign engineers to identify the ways to increase the expected outcomes beforehand. The current model provides predictions for Open rate, Click-to-open rate, and Conversion rate. The campaign engineer can upload the email with the discount for the campaign and select the industry, campaign type, and preferred range of discounts as shown in figure 2.



Please select your industry

Retail

Please select your campaign type

Promotional

Please select the target varible you want to optimize

Conversion_Rate

Please select the range of discounts you prefer

70 - 80

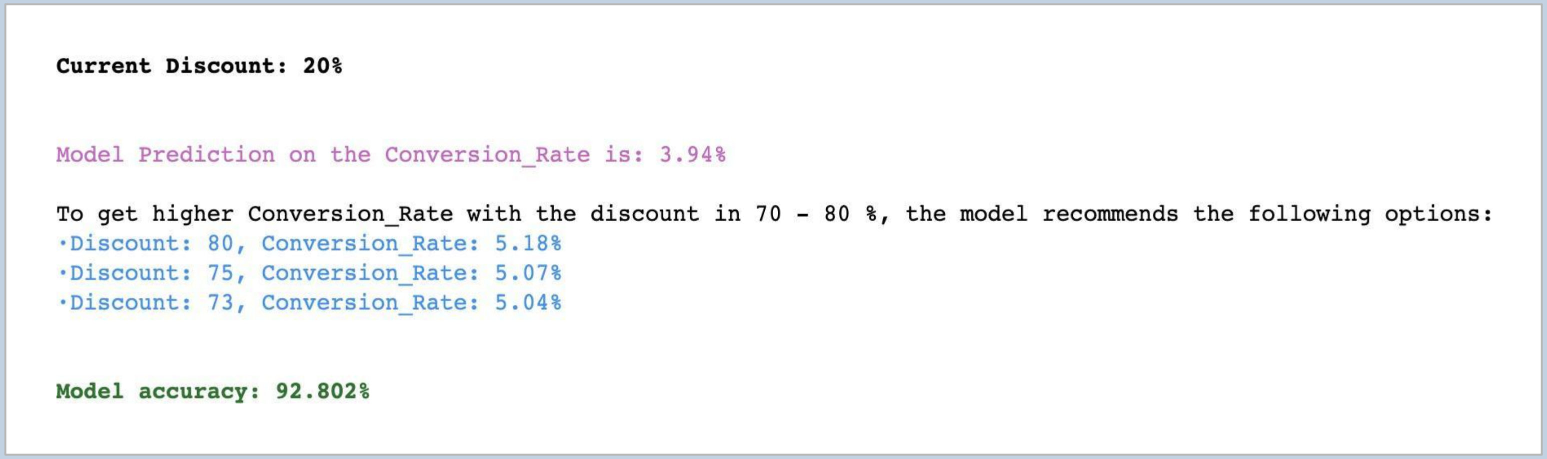
Figure 2. Set of parameters to be selected by the user's preference for the model

After selecting the above parameters, the model will run for the given email to provide predictions. It will provide the predicted rate, recommendations to increase the rates while having the discount value in the desired range, and the model prediction accuracy. Figure 3 shows a sample output. Based on the model output, the campaign engineer is capable of deciding the most suitable discount value to be used within their desired range to get the most engagement rates. Using the discount model, the campaign engineer can foresee the outcome of the campaign and take actions to increase the outcomes.

IX. MODEL VALIDATION

For machine learning, XGBoost was selected after comparison with other machine learning algorithms. In our experiments, we used random forest, decision tree, linear re- gression, and kernel ridge regression (with polynomial/non- linear kernel). The results are depicted in table 2. According to the table, it is apparent that the random forest and XGBoost accuracies are at the highest level.

As shown in figure 4, their accuracies are much closer to each other and for open rate and CTO rate, Random forest has better accuracy. But in order to select the best among the two algorithms, their runtimes were also compared. Figure 5 depicts their runtimes. It is clear from the runtimes that xgboost has a much lower runtime (closer to 6x less) compared to random forest. Therefore, xgboost is able to provide the output to the user within a few seconds.

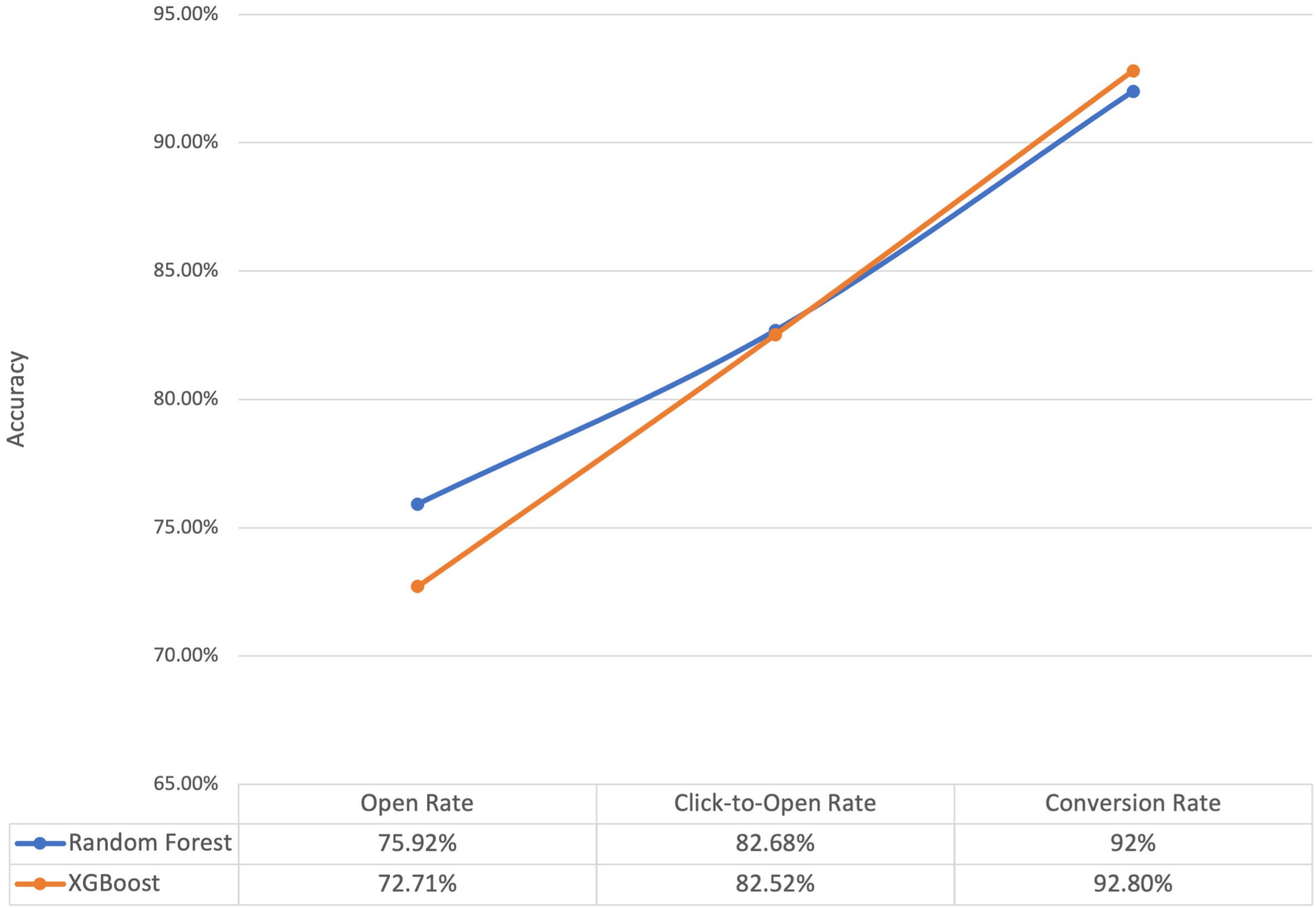


```
Current Discount: 20%


Model Prediction on the Conversion_Rate is: 3.94%

To get higher Conversion_Rate with the discount in 70 - 80 %, the model recommends the following options:
•Discount: 80, Conversion_Rate: 5.18%
•Discount: 75, Conversion_Rate: 5.07%
•Discount: 73, Conversion_Rate: 5.04%


Model accuracy: 92.802%
```

Figure 3. Discount model output with predictions and recommendations

	Random Forest	XGBoost	Decision Tree	Linear Reg.	Kernel Ridge
Open Rate	75.92%	72.71%	45.22%	6.88%	13.53%
CTO Rate	82.68%	82.52%	81.50%	8.98%	5.95%
Conversion Rate	92%	92.80%	87.01%	30.02%	6.21%

Table 2. Accuracy values with different machine learning models for the target variables, Open Rate, Click-to-Open Rate (CTO Rate) and Conversion Rate



	Open Rate	Click-to-Open Rate	Conversion Rate
Random Forest	75.92%	82.68%	92%
XGBoost	72.71%	82.52%	92.80%

Figure 4. Accuracy comparison between Random Forest and XGBoost algorithms

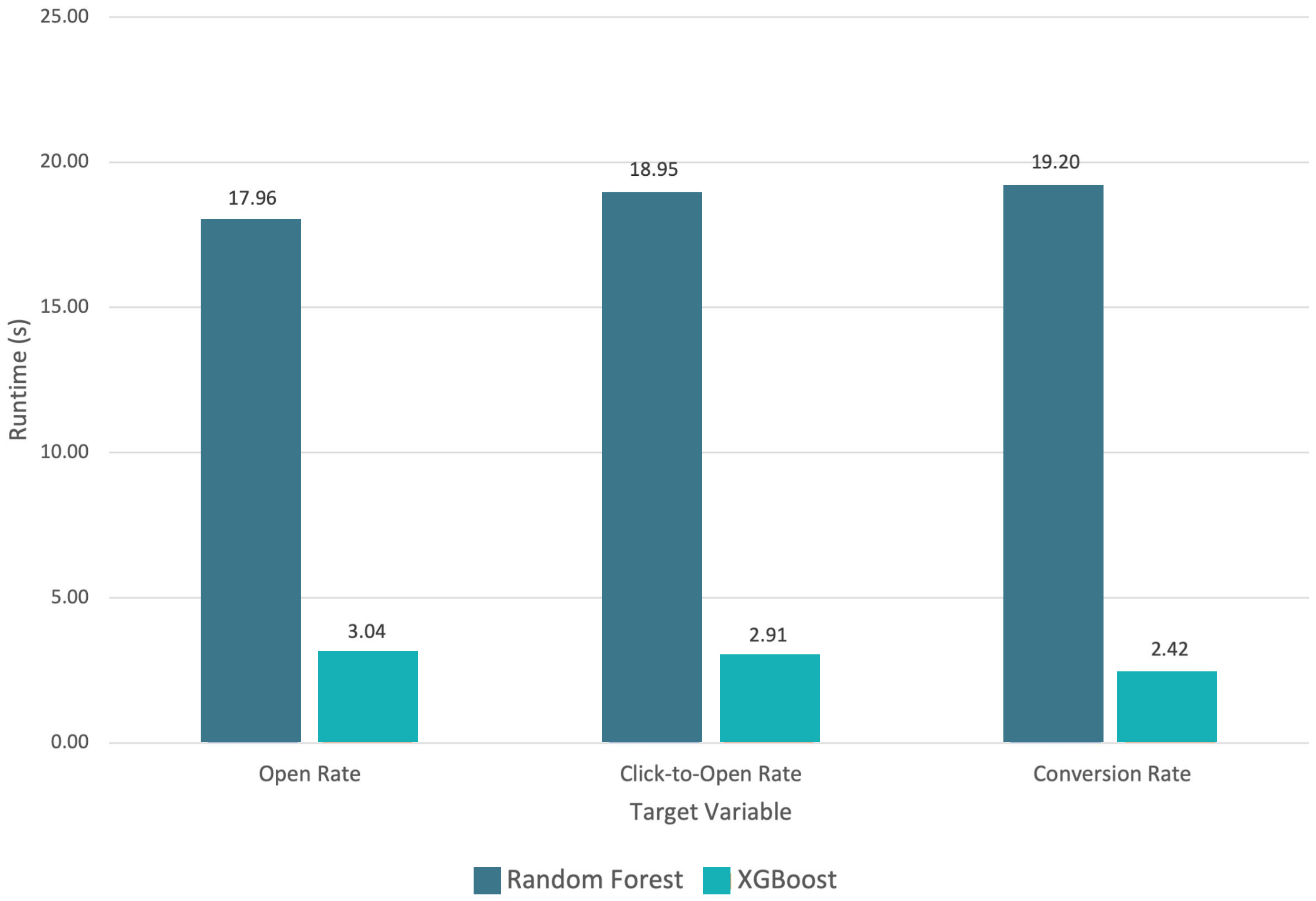


Figure 5. Runtime comparison between Random Forest and XGBoost algorithms

X. ASSUMPTIONS

There are a few assumptions that had to be considered during the model-building process. When multiple discounts occur in the email, the model considers the highest number as the discount for the model predictions. Due to the lack of availability of data for the model, it refers to the email benchmark data, and assumptions are made to decide on the range of the distributions of target variables based on the average benchmarks. Also, the values are assigned with the assumption that the higher the discount, the higher the engagement.

XI. DATA SCIENTIST'S OWN CONCLUSIONS

The discount optimization model gives predictive analytics for the discount email campaigns by giving recommendations to improve outcomes within a few seconds. According to the results from the discount optimization model, it is able to provide higher accuracies for the target variables using the current dataset with 1183 emails. This shows that the model can identify the trends related to the discount email campaigns and correctly predict outcomes.

For the conversion rate, the model provides 92.8% accuracy which validates the performance of the discount optimization model. The accuracy values for all three targets are above 70%, showing the potential of the model to be used in real-world scenarios. As these results are based on benchmark data, the model will gain further information and gain more performance when used in real-time.

XⅡ. FUTURE CONSIDERATIONS

For immediate future work, the model will be extended to predict *Revenue-per-email* values and also to recommend the discount value to get the highest possible rate (not only within the discount range selected). More email samples will be added to the dataset to improve recommendations. Also, the model will be adjusted to better handle multiple discounts in a single email than selecting the highest discount. To expand the use of the model, it will be modified to consider discounts in both percentages and '$' amounts as well. The current model is not compatible with image-only emails and it will be soon implemented to work with image emails as well. As the current machine learning model uses default hyperparameters, a hyperparameter tuning process will be done to further improve the performance.

References

[1] 2016. Ecommerce Email Marketing Benchmarks. https://cm- commerce.com/academy/email-marketing-benchmarks/

[2] 2018. Email Conversion Rate Benchmarks. https://www.listrak.com/ white-papers/2018-email-benchmarks

[3] 2021. Ultimate Email Marketing Benchmarks for 2021: By Industry and Day. https://www.campaignmonitor.com/resources/guides/email- marketing-benchmarks/

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[5] Katie Holmes. 2021. Average Conversion Rate by Industry and Marketing Source. https://www.ruleranalytics.com/blog/insight/conversion- rate-by-industry/

[6] Ameet V Joshi. 2020. Amazon's Machine Learning Toolkit: Sagemaker. In *Machine Learning and Artificial Intelligence*. Springer Nature, Chapter 24, 233–243. https://doi.org/10.1007/978-3-030-26622-6_24

[7] Yanli Liu, Yourong Wang, and Jian Zhang. 2012. New machine learning algorithm: Random forest. In *International Conference on Information Computing and Applications*. Springer, 246–252.

[8] Enricko Lukman. 2021. What is a good conversion rate for your business? https://www.contentgrip.com/conversion-rate-business-benchmark/

[9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. http://scikit-learn.sourceforge.net.

[10] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).