

Data-Driven Strategies for Improving Email Campaign Engagement: A Send Time Optimization Approach

Dalia Saleh Abbas, Mustafa Al-Jailawi

Elektroteknik
Datateknik

LU-CS-EX: 2024-17

DEPARTMENT OF COMPUTER SCIENCE

LTH | LUND UNIVERSITY



EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2024-17

**Data-Driven Strategies for Improving
Email Campaign Engagement: A Send
Time Optimization Approach**

Datadrivna metoder för att öka
engagemanget i e-postkampanjer:
Optimering av sändningstider

Dalia Saleh Abbas, Mustafa Al-Jailawi

Data-Driven Strategies for Improving Email Campaign Engagement: A Send Time Optimization Approach

Dalia Saleh Abbas
da2447sa-s@student.lu.se

Mustafa Al-Jailawi
mu5823al-s@student.lu.se

September 11, 2024

Master's thesis work carried out at IKEA.

Supervisors: Fredrik Danielsson (INGKA), fredrik.danielsson1@ingka.ikea.com
Pontus Giselsson (LU), pontusg@control.lth.se

Examiner: Jacek Malec, Jacek.Malec@cs.lth.se

Abstract

Email direct marketing is one of the cornerstone pillars of digital marketing playing a pivotal role in engaging target audiences and achieving campaign success.

This collaborative research between IKEA and Lund University explores optimizing and tailoring email send times to better correspond with customers' behavioral patterns. The study delves into the realm of send time optimization (STO), a methodology marketers adopt to enhance the impact of email campaigns by increasing customer engagement. Furthermore, this study builds upon prior research by incorporating web engagements alongside Recency, Frequency, and Monetary (RFM) values as additional features. The incorporation of web engagements not only facilitated tailored send times to a wider range of customers but also enhanced the scope of data collected about each customer.

Through the use of various regression algorithms, the model was trained on data from 90 days of customer interactions. The evaluation results indicated XGBoost to be the best-performing algorithm on the dataset, leading to its selection for A/B testing in a production environment. The A/B test, conducted over two weeks, was designed to benchmark the newly proposed model against the current model at IKEA. The comparative analysis evaluates the efficacy of each model by examining the click-through, open, and opt-out rates. The findings suggest marginally superior performance exhibited by the proposed model compared to IKEA's current model. Notably, both models outperform fallback send times, indicating that optimizing send times positively influences email engagement.

Keywords: Send time Optimization, STO, Regression, Increase engagement, Send Hour, Marketing Emails, Subscriber

Acknowledgements

First and foremost we would like to thank IKEA for providing us with the equipment, data and a network of highly skilled people. We would especially like to thank our supervisor at IKEA, Fredrik Danielsson for providing valuable insights and connecting us with the right people. Moreover, we appreciate everyone at IKEA who took the time to meet with us and brainstorm ideas.

Additionally, we would like to express our sincere gratitude to Francois Domain, Andrew Bradley, along with the rest of the team at VML MAP for their invaluable assistance in establishing the testing environment, which provided us with the opportunity to assess the performance of our model in real-world scenarios. Further, we thank our LTH supervisor, Pontus Giselsson, for guiding us through the thesis. Lastly, a special thank you to everyone who interacted with IKEA emails, without you this thesis would not be possible.

Contents

1	Introduction	7
1.1	Thesis Purpose	8
1.2	Research Questions	8
1.3	GDPR	9
2	Background	11
2.1	Related Work	11
2.1.1	Classification Approach	11
2.1.2	Regression Approach	12
2.2	STO Model Currently Used at IKEA	12
2.3	Limitations	13
3	Theory	15
3.1	Machine Learning Methods	15
3.1.1	Classification	15
3.1.2	Regression	16
3.2	Machine Learning Algorithms	19
3.2.1	XGBOOST	19
3.2.2	K-means	20
3.3	Categorical Encoding Methods	21
3.3.1	Label Encoding	21
3.3.2	One-Hot Encoding	21
3.4	Overview of A/B Testing	22
3.5	Email Metrics for A/B Test Evaluation	23
3.5.1	Open-rate	23
3.5.2	Click-through rate	23
3.5.3	Opt-out-rate	24
3.6	Email Types	24

4	Method	25
4.1	Methodological Alignment Across Models	25
4.1.1	Email Type Selection	25
4.1.2	Data Preprocessing	26
4.2	Model 1-Historical Interaction-Based Model	27
4.2.1	Model 1.1	28
4.2.2	Model 1.2	28
4.3	Model 2-Demographics-Integrated Model	29
4.4	Model 3-Customer Engagement Prediction Model	31
4.4.1	Classification Approach	33
4.5	Methodology of A/B Testing the Model	34
5	Model Evaluation and Discussion	35
5.1	Model 1-Historical Interaction-Based Mode	35
5.2	Model 2-Demographics-Integrated Model	36
5.3	Model 3-Customer Engagement Prediction Model	37
5.3.1	Time Intervals	38
5.3.2	Web Data	39
5.3.3	Predictions	39
5.3.4	Data Engineering	40
5.3.5	Binary Classification	41
5.3.6	24-label Classification	42
5.3.7	A/B Testing	43
5.4	Future Work	43
6	Conclusion	45
	References	47

Chapter 1

Introduction

In the modern digital landscape, email communication has become an essential tool for businesses to reach out to consumers. Just in 2022 alone, 319 billion business and consumer emails were sent and received daily, projecting an increase to 376 billion emails being sent and received in 2026 [5]. Email communication is a cost-effective strategy that involves sending messages to customers for order confirmations, digital receipts, promoting products and services, as well as building a relationship with customers, just to name a few. It is a direct and targeted type of communication that allows businesses to reach customers in a personalized fashion.

Digital email marketing plays a pivotal and cost-effective role in contemporary business communication. One remarkable metric highlighting its effectiveness is the Return On Investment (ROI). Specifically, in email marketing, every 1 USD invested yields an impressive average return of 36 dollars. In certain sectors like retail and e-commerce, this figure can soar even higher, reaching up to 45 USD [5]. This highlights the significance of leveraging digital marketing strategies, particularly email campaigns, as they not only facilitate effective communication but also demonstrate a substantial return on investment for companies.

Despite the cost-effectiveness and impressive ROI associated with email marketing, the challenge arises from the sheer volume of emails sent daily to billions of consumers, leading to issues such as emails being lost in the inbox, flagged by spam filters, or outright ignored. The primary culprit behind these challenges is the overload of emails, where individuals receive a multitude of messages on a daily basis, making it easy for emails to get lost or overlooked in the deluge of communication. Several contributing factors exacerbate this problem such as irrelevant content or the delivery of emails at an unwanted frequency and inconvenient timings can lead recipients to ignore messages that fail to deliver genuine value, and in adverse cases, prompt customers to opt out of further communications.

1.1 Thesis Purpose

This paper seeks to delve into solutions aimed at enhancing email engagement by specifically addressing the issue of email dispatch timings. Given that messages often get lost or overlooked when sent during busy periods or non-optimal times of the day, this study adopts a Send Time Optimization (STO) approach. The focus is on tailoring email send times to align with the hours when customers are most likely to actively engage with their inbox. While STO can encompass factors such as email content, subject lines, and relevance for individual consumers, this paper places particular emphasis on the timing of email dispatch as a key solution to bolstering email consumer engagement.

The approach of this thesis is specifically designed to align with the preferences and requirements within the IKEA ecosystem. The process involves the incorporation of STO into a targeted advertising approach for email marketing. In this context, emails are scheduled for dispatch without relying on the model's capability to determine the relevance of the email content for a customer. Given that existing systems are adept at ascertaining the relevance of email content for recipients, the primary emphasis is placed on optimizing the timing of email dispatch.

The secondary objective of this thesis is to conduct a comparative evaluation between the thesis-produced STO model in collaboration with Lund University and an STO model currently utilized by IKEA for optimizing the dispatch timings of marketing emails. This evaluation will be conducted within the context of a single campaign and market to determine if the thesis model performs at or above an industry standard level. If successful, the model could be adapted for broader application across various aspects of IKEA's email communications, thereby supporting more personalized and timely email marketing initiatives at IKEA.

1.2 Research Questions

- **Question 1:** How does the thesis-developed Send Time Optimization model measure up to the Send-Time Optimization model that is currently adopted by IKEA?
- **Question 2:** How do we effectively manage email delivery scheduling for customers with no prior engagement data in the Send Time Optimization model? What strategies or methodologies can be employed to ensure optimal email send-out times for these subscribers, considering the absence of historical interaction patterns?
- **Question 3:** What specific Send Time Optimization strategies have proven effective in enhancing email engagement?

1.3 GDPR

INGKA is an international enterprise that owns and operates IKEA stores worldwide. It manages the retail operation of IKEA stores under franchise agreements with Inter IKEA Systems B.V.

INGKA is dedicated to upholding rigorous data privacy standards and complying with the provisions of the General Data Protection Regulation (GDPR). The research collaboration between Lund University and INGKA strictly operates within a robust framework of data ethics, respecting user privacy and confidentiality. It is crucial to note that all data used in this study is sourced and processed in strict accordance with principles of the GDPR, with a formal agreement in place to define data sharing. Throughout this thesis, we prioritize individual privacy rights and ethical data handling, ensuring fairness, transparency, and accountability.

Chapter 2

Background

The following chapter highlights relevant and related work found in literature including a brief background about the STO model currently adopted at IKEA. This chapter further explores the limitations of this paper.

2.1 Related Work

This section highlights different modeling approaches to STO that have been found in the literature. The predictive approaches to the problem have been divided into two branches; classification and regression. The models in the literature are based on customer profile data in conjunction with customer behavior data, such as open and click-through rates.

2.1.1 Classification Approach

Paralic et al. [15] aimed to improve the open rate of emails sent to customers in marketing campaigns. The model uses classification into 3 phases (i) a model that classifies whether a user opens or ignores an email and thus filters out users who are predicted to not open the email and will therefore not be considered in a particular campaign. If a user is classified to open the email they will continue to (ii) a 24-label classifier to predict the best hour of the day and (iii) a 7-label classifier that predicts the preferred day of the week. Paraliac et al. achieved an accuracy of 80.54% using a decision tree for predicting the best hour of the day. The study concludes that the model should increase the open rate on the site of potential customers. Further work is to run the experiment in a live environment with a cooperating company.

Luo et al. [14] aimed to improve open rates using two classification algorithms by modeling whether a subscriber will open or ignore an email based on email content, historic email interaction information such as the day and time when the email was opened, and demographic data. The study used Decision Trees and Support Vector Machines, both based on

email and customer profile data. The Decision Trees with an F1-score of 80% on the open predictions outperformed Support Vector Machines with an F1-score of 74% on open predictions.

What both studies have in common is the goal of maximizing the open rates and thus excluding the subscribers that the model predicts to be non-openers for a certain campaign or email.

2.1.2 Regression Approach

Deligiannis et al. [7] aimed to evaluate a new method for scheduling reminder messages compared to a company's standard practice of fixed time intervals. The evaluation focused on message seen and click-through rates, as well as customer repurchase rates. The new optimized scheduling was tested on a cohort of 40 repeat customers, while historical data from a year earlier formed a benchmark cohort under the previous standard method.

Results from the one-week live deployment indicated that the new method outperformed the standard approach. The experiment cohort consistently showed higher message seen and click-through rates, as well as increased customer repurchase rates at 1, 2, and 24 hours. Statistical significance was observed with a 95% confidence level. A comparison of repurchase rates over time illustrated the superior performance with optimized reminders. Overall, the findings supported the hypothesis that the optimized scheduling method led to higher customer engagement and repurchase rates compared to fixed time intervals.

Limitations of this project are the short testing period as well as the small size of the dataset. Also worth noting is that the study focused more on what day to schedule the message and less on what hour of the day to send the message.

Araújo et al. [2] introduces a service that estimates the optimal day of the week as well as the time of the day for sending a marketing message to individual subscribers. The project segments subscribers using profile attributes, such as equipment, email domain, operating system and lastly demographic data. The paper [2] uses regression to predict the optimal period to send a marketing message and experiments with various regression algorithms are done with the K-Neighbors Regressor coming on top with an R^2 of 0.898 and MAE of 0.166 (see section 3.1.2 for further explanation of regression metrics). The study further creates an ML Ensemble model stocking multiple regressors achieving an R^2 of 0.91 and MAE of 0.204. The paper proposes future work to segment subscribers into recency, frequency, and monetary value (RFM) as well as further segmentation of the lifetime value (LTV) of subscribers.

The research outlined in this paper employs a regression approach, distinguishing itself from prior studies along three critical dimensions. This project introduces a model that not only (i) forecasts the optimal send hour without excluding potential nonresponders but also (ii) incorporates subscribers' RFM values. Furthermore, (iii) it introduces a novel layer of web data interactions to encompass a larger dataset of subscribers' interaction behaviors.

2.2 STO Model Currently Used at IKEA

The current model adopted by IKEA is an AI model that uses 90 days of email engagement data to train the model and determine the best time to send an email to each customer within a 24-hour interval. The idea behind it is to analyze individual customer engagement data

such as open and click rates and through that predict the most optimal time. The model is essentially a black box and little to no information is known about the details of its implementation.

2.3 Limitations

While our study highlights valuable insights into the development and usage of an STO model in a business context, it is important to highlight the limitations of the study that might have impacted the final results of the model and our findings. Pointing out these limitations is a crucial part of ensuring the reproducibility of the work and guiding future research in this area.

Data is one of the most significant components of any machine learning model, and the quality of the data determines the quality of the model. A considerable limitation is missing or incomplete data, which was a frequent challenge in the preprocessing stage of the study. Upon studying a sample dataset consisting of the features needed for the model, we found that 40% had at least one missing piece of information. These issues required careful evaluation and handling to avoid introducing noise into our analysis.

Furthermore, an important factor to take into account when working with customer-based data is that biases can arise. This can occur in various ways, such as through the data collection process, the sampling methods, or inherent biases in the studied customer sample. The model was trained using 90 days of data sourced from customers residing in one specific country who have engaged with IKEA, with weekly training sessions conducted. There were a number of biases introduced by this method of working, including disproportionate representation and an inability to account for seasonality or external factors such as holidays and promotions.

Moreover, when attempting to create a model that identifies patterns in human behavior, it is necessary to consider various factors that drive these behaviors. Human behavior is extremely complex, so there can be no guarantee that there is always a clear pattern or connection between different factors. A model created for these predictions needs to account for this complexity and what drives certain behaviors from perspectives such as psychology, sociology, anthropology, and neuroscience. However, these types of data are heavily regulated and scarce, which presents a limitation when trying to factor in behavior-related features within companies like IKEA.

Lastly, one notable constraint we encountered during the model testing phase was time limitations. Achieving statistically significant evaluation results necessitated running the model for an extended duration. However, due to the time constraints inherent in the thesis project, we were unable to allocate the ideal timeframe required for optimal results.

Chapter 3

Theory

In our empiric study certain theoretical frameworks served as the fundamental building blocks. As part of this chapter, we aim to introduce the key concepts underlying these building blocks by providing an overview of the algorithms used, the evaluation process and an overview of the various types of email types and their influence on the construction of the model.

3.1 Machine Learning Methods

The nature of the learning task that machine learning algorithms are designed to carry out is what broadly categorizes them into different types. In this section, we explore two fundamental machine learning types: classification and regression.

3.1.1 Classification

The fundamental objective of a classification model is to correctly predict the class label based on the features. Typical classification techniques include logistic regression, decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and neural networks [3]. The implementation approaches and methodologies used in each of these approaches vary in degree of complexity and performance, but they all share the fundamental concept of categorizing input data into categories.

Various methods can be used to assess a classification model. Metrics commonly used include:

- **Accuracy:** The classifier's prediction accuracy percent. Essentially accuracy is the ratio between the number of correctly classified predictions and the total number of predictions. It can be described with the following formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (3.1)$$

where TP are the true positives, TN are the true negatives, FP are the false positives and FN the false negatives

- **Precision:** In the case where False Positive is of higher concern than False Negatives, we can calculate precision. It is essentially the ratio between the True Positives and the total number of positives. It can be described with the following formula:

$$Precision = \frac{TP}{TP+FP}. \quad (3.2)$$

- **Recall:** Recall is a metric used to measure how well the model can correctly identify positive instances from the samples that are actually positive in the dataset. It can be described with the following formula:

$$Recall = \frac{TP}{TP+FN}. \quad (3.3)$$

A high recall indicates that the model is good at finding the positive instances of the target in the dataset. A recall of 1.0 indicates that the model is able to find all instances.

- **F1-score:** The F1-score calculates the harmonic mean of precision and recall. Essentially, a high F1-score indicates that the model performed well on both the recall and the precision.

It can be described with the following formula:

$$F1 = 2 \cdot \frac{Precision+Recall}{Precision \cdot Recall}. \quad (3.4)$$

- **AUC-ROC:** AUC also known as Area Under the Curve, is a metric used in order to measure the classifiers ability to separate between classes. A ROC curve plot has an y-axis that plots the true positives and an x-axis that plots the false positives. The area under the graph gives us the AUC-ROC, where a score of 1 indicates perfect performance of a binary classification model. A diagonal line starting from 0 has an AUC-ROC of 0.5, indicating that the model has no ability to separate the binary classes and is as good as a random guess.

The definitions have been taken from the following sources: [6] [17]

3.1.2 Regression

A regression model's objective is to fit a mathematical function to independent variables (features) and dependent variables (targets) in order to establish a link between the two. Common regression techniques include Linear Regression, Decision Tree Regression, Random Forest Regression, and algorithms such as Gradient Boosting and XGBoost. Gradient Boosting is an ensemble learning technique that builds a strong model by sequentially adding weak learners, each correcting the errors of its predecessor. XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting algorithms. While Gradient Boosting and XGBoost are versatile and can be utilized for regression tasks, they are not inherently exclusive to regression. Each of these methods uses a different strategy to predict a continuous value based on one or more input characteristics.

- **Linear Regression:** In linear regression it is assumed that the relationship between the dependent and independent variables are linear. The idea is for the model to fit a linear line to the data points. The linear line has the following formula:

$$Y = a + bX + u, \quad (3.5)$$

where a is the intercept, which indicates the baseline of the dependent variable, b is the slope coefficient and u is the error term.

The coefficients a and b are calculated using a method called Ordinary Least Square, the goal is to find the values of a and b that minimizes the sum of squared differences between the observed values of the dependent variable and the values predicted by the regression model [18] [1].

- **Decision Tree Regression:** In the Decision Tree Regression the relationship between features and target is modeled by recursively partitioning the feature space into sections and then assigning each section a constant value. Based on these values and the mean square errors of the splits the tree is either split to the left or the right until a final leaf node is reached [16].

Ensemble techniques

The following regression approaches are based on Ensemble learning. The idea behind it is to create multiple models often using a weaker regression model and then combining them in order to improve the predictive performance [20].

- **Random Forest Regression:** Random forest builds multiple Decision Trees and combines them in order to make the finale prediction [20].
- **Gradient Boosting Regression:** Gradient Boosting Regression builds an ensemble of decision trees sequentially. The idea behind it is that a learner corrects the errors made by the previous ones and thus minimizing the error for each step. Once all the trees are trained the predictions are combined to make the finale prediction [22] .
- **XGBoost Regression:** Extreme Gradient Boosting (XGBoost) regression is an optimized version of gradient boosting regression. The difference is that XGBoost incorporates advanced regularization techniques and tree pruning to improve efficiency and predictive performance [4].

The regression models above are all evaluated with the same metrics, commonly used metrics to evaluate and compare the performance of a regression model are the following:

- **Mean Absolute Error (MAE):** MAE measures the difference between the predicted value and the actual value. The value is obtained by taking the absolute value of the difference between them for each data point and then calculating the average. It can be described with the following formula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.6)$$

where y_i is the actual output value, $\hat{y}_i$ is the predicted output value and n is the total number of data points.

A smaller MAE suggests that the model is more accurate in predicting the actual values [17].

- **Mean Squared Error (MSE):** MSE measures the square difference between the predicted value and the actual value. It is computed by squaring the difference for each data point, and then finding the average of these squared differences.

It can be described with the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (3.7)$$

Because of its property of squaring the differences, MSE has the advantage of prioritizing larger errors over smaller ones. A smaller MSE suggests that the model is more accurate in predicting the actual values [17].

- **Median Absolute Error (MedAE):** MedAE is very similar to MAE, with one significant distinction: it computes the median instead of the mean absolute error over the calculate absolute differences. This characteristic renders the metric less sensitive to outliers. It can be described with the following formula:

$$MedAE = median(|y_i - \hat{y}_i|). \quad (3.8)$$

A smaller MedAE suggests that the model is more accurate in predicting the actual values [12].

- **Mean Squared Logarithmic Error (MSLE):** MSLE measures the difference between the predicted and actual values that have been transformed into logarithmic units. It is computed by taking the mean of the squared differences between the logarithms of the actual and predicted values, each incremented by 1, across all samples in the dataset. It can be described with the following formula:

$$MSLE = \frac{1}{n} \sum_{i=1}^n (\log(y_i + 1) - \log(\hat{y}_i + 1))^2. \quad (3.9)$$

A smaller MSLE suggests that the model is more accurate in predicting the actual values [10].

- **R-squared :** Statistically, R-square measures the extent to which the independent variables contribute to the variation of the dependent variables. R-squared is computed through a two-step process. First, the unexplained variation is calculated by summing the squared errors, similar to the methodology used in calculating MSE. Then, this sum is divided by the total variation, which is calculated by adding up the differences between each actual value and the average of all actual values. The result is then subtracted from one to obtain the R-squared coefficient. It can be described with the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (3.10)$$

where $\bar{y}$ is the mean of the actual values. The R-squared value ranges between 0 and 1. A score of 1 signifies that all variations in the dependent variable can be explained by variations in the independent variable, implying better predictive performance as the score approaches 1 [9].

3.2 Machine Learning Algorithms

The choice of the right machine learning algorithm is of critical importance to the success of the model. In our study, we used two widely used machine learning algorithms: XGBoost and k-means clustering.

3.2.1 XGBOOST

XGBoost, short for eXtreme Gradient Boosting is a particularly powerful implementation of gradient boosting algorithms and could be used for building supervised regression models. This method involves combining weak learners, such as decision trees, iteratively in order to create a strong learner, which is known as boosting. XGBoost offers numerous advantages over other machine learning algorithms and consistently outperforms them [4]. The model can be described with the following algorithm [19]:

Algorithm 1 XGBoost Algorithm

Data: Input data X , Regularization parameter λ , Learning rate ρ , Maximum depth max_depth , Threshold γ , Validation set X_{val}

Result: Output result Y

Initialize a single leaf tree

tree depth $\leftarrow 0$

while $tree_depth \leq max_depth$ **do**

 Make initial predictions by averaging the target variable values

 Compute residuals using the chosen loss function

 Calculate the root similarity score:

$$\text{Root Similarity_Score} = \frac{(\sum \text{Residuals})^2}{\text{Number of Residuals} + \lambda}$$

 where λ is the regularization hyperparameter

end

foreach *possible split* **do**

 Compute left similarity score and right similarity score:

$$\text{Left Similarity_Score} = \frac{(\sum \text{Left Residuals})^2}{\text{Number of Left Residuals} + \lambda}$$

$$\text{Right Similarity_Score} = \frac{(\sum \text{Right Residuals})^2}{\text{Number of Right Residuals} + \lambda}$$

 Compute the information gain:

$$\text{Gain} = \text{Left Similarity Score} + \text{Right Similarity Score} - \text{Root Similarity Score}$$

if $\text{Gain} > \text{threshold } \gamma$ **then**

 Split the current node

Update the residuals for the next iteration:

$$\text{New residuals} = \text{Old Residuals} + \rho \cdot \Sigma \text{Predicted Residuals}$$

where ρ is the learning rate

Increment the tree depth: $\text{tree depth} \leftarrow \text{tree depth} + 1$

The initial step in the XGBoost algorithm is to establish a baseline model. This baseline is constructed by making an initial prediction. Typically, this prediction is computed by taking the average of all the target variables. Visualizing this as a line, we then assess the difference between each actual target value and this baseline prediction, termed residuals. These residuals essentially quantify the distance between the target variable and the initial model.

Then, the XGBoost algorithm iteratively refines this base model to minimize these residuals. By fitting subsequent models to these residuals, the idea behind XGBoost is to progressively improve predictions. This is achieved by evaluating potential splits in the data using a similarity score. Using the similarity score for the root, and right and left node, we then calculate the gain, which represents the improvement obtained from each split. Following that, the root is either split or not split based on a predefined threshold. If the gain is higher than the threshold, which indicates improvements in the split, the split is executed; otherwise, the process halts, maintaining the existing structure.

The model's predictions from the current iteration are then scaled by the learning rate and then added to the previous residuals. This update mechanism helps adjust the residuals based on the model's latest predictions, guiding the model towards better predictions in subsequent iteration.

This iterative process continues until the maximum tree depth is met. By iteratively refining the model's structure and minimizing residuals, XGBoost effectively utilizes ensemble learning to make accurate predictions.

3.2.2 K-means

K-means is an unsupervised machine learning algorithm. The goal of the algorithm is to group similar data points together into k clusters based on the underlying patterns of the

data points, thus creating clusters that contain data points that have certain similarities. K-means can be described with the following algorithm [21]:

Algorithm 2 K-means Algorithm

Data: Input data X , Number of clusters k

Result: Set of k clusters

Randomly initialize k centroids $C = c_1, c_2, \dots, c_k$;

while not converged **do**

foreach data point $x_i \in X$ **do**

Calculate the distance between x_i and each centroid c_j ; Assign x_i to the nearest centroid: $y_i = \arg \min_j \|x_i - c_j\|^2$;

foreach centroid c_j **do**

Update its location as the mean of all points assigned to it: $c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$, where S_j is the set of points assigned to c_j ;

Check for convergence by comparing the new centroids to the previous centroids;

3.3 Categorical Encoding Methods

In general, machine learning algorithms can only handle numerical inputs, but categorical inputs may be needed in certain situations. Categorical variables represent characteristics or attributes with discrete values. In this part, we look at two common encoding techniques: label encoding and one-hot encoding. These strategies enable us to convert categorical data into a format usable by machine learning algorithms.

3.3.1 Label Encoding

Label encoding transforms categorical variables by assigning a unique integer value to each category. The objective of this encoding is to preserve the ordinal relationship among the categories by representing them with ordered numerical values. For instance, if we were to encode income ranges such as "Low", "Medium", "High" where there exists a natural order, the encoding would assign them the numerical values 0, 1 and 2, respectively.

An advantage of using label encoding is its simplicity and ability to preserve ordinal information. However, if used on data that is not appropriate for this type of encoding, it may introduce unintended ordinality into nominal variables, which could have a negative impact on the machine learning algorithm's performance.

An important consideration before applying Label Encoding is to carefully study the nature of the categorical variables and decide whether or not it is appropriate to introduce ordinality [13].

3.3.2 One-Hot Encoding

One-hot encoding encodes categorical data by transforming it into binary form. Essentially, this requires constructing as many binary columns as there are distinct categories; each column will then represent a category, with just one column having a value of 1, indicating the

presence of that category. For example, if we have the colours "RED", "GREEN", and "BLUE", three binary columns will be formed. In the first category, just the first column will have a 1, while the remaining columns will be 0. The same principle is then applied to the remaining two groups.

This sort of encoding has the benefit of being compatible with a wide range of machine learning algorithms and being able to handle categorical variables in many categories. However, if the data contains many distinct categories, utilizing one-hot encoding will result in the curse of dimensionality due to the generation of a huge number of additional features. This increase in dimensionality might provide challenges such as increased computing complexity, sparsity, overfitting, and difficulty interpreting and understanding the model's behavior.

It is important to consider the difficulties and additional complexity that One-hot encoding brings to the model while encoding data. To manage the increase in data size and complexity, consider using feature selection techniques [13].

3.4 Overview of A/B Testing

The A/B testing methodology, also known as split testing, is used when conducting customer experience research. It is essentially a randomized experiment where different versions of the same component are tried concurrently to see which is more effective in yielding specific results depending on the context of the experiment. A/B testing entails segmenting users into equally large, separate subgroups and then implementing the various versions that should be tested on each segment [8].

When incorporating the STO model into email marketing campaigns, the experimental variations are compared based on click rate, click-through-rate, and opt-out-rate to determine which version produces optimal outcomes.

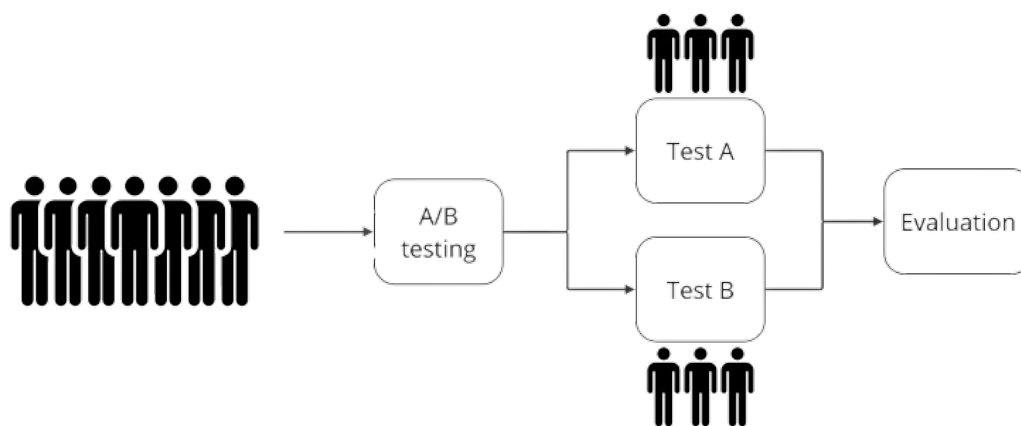


Figure 3.1: A/B testing split.

Overall, A/B testing gives us valuable insights into the actions of the clients receiving the emails, allowing us to make data-driven decisions about how to increase the efficacy of the email marketing campaigns.

3.5 Email Metrics for A/B Test Evaluation

An important component of this study is to systematically evaluate the STO model's performance through real-time customer data in order to get the full understanding of the efficiency and impact of the model on customer engagement. The performance is quantified using metrics such as click-through rate, open-rate and opt-out-rate. These metrics provides valuable insights and perspectives on the application of an STO model on emails.

3.5.1 Open-rate

The open rate is a measure that indicates how many of the emails sent are opened. This measure is a crucial indicator of the recipient's interaction with the email that was delivered.

The open rate is calculated by dividing the number of opens by the number of emails sent and then multiplying by 100 to get the percentage.

$$Open_rate = \frac{NumberofOpens}{NumberofDeliveredEmails} \cdot 100\% \quad (3.11)$$

In the context of STO, including open-rate in the model evaluation step provides us with a good first indication of the model's efficacy and whether a personalized send time has an impact on capturing the customer's attention and driving them to open the email. One drawback of this metric is that in certain cases, such as with Apple Mail, emails are opened automatically due to the application's feature designed to enhance user experience by providing a quick glimpse of the email's content without requiring the user to explicitly open the email. This behavior can potentially skew the results and introduce bias in tracking email open rates. Fortunately, this automatic opening accounts for only a small percentage of the emails [11].

3.5.2 Click-through rate

One of the most important metrics in our evaluation is click-through rate (CTR), which measures not only the effectiveness of recipients opening their emails but also the effectiveness of driving a person to take action when an email is opened. Essentially, it represents the percentage of customers who have opened an email and interacted further with that email by engaging with one or more links in the message. In A/B testing, this metric is critical to understand how the model and the prediction of an optimal time drive actions such as making a purchase, visiting the website, or responding to the survey.

The CTR is calculated by dividing the number of clicks on links in sent emails by the number of emails sent and multiplying by 100 to get the percentage.

$$CTR = \frac{Number\ of\ Clicks}{Number\ of\ Delivered\ Emails} \cdot 100\% \quad (3.12)$$

In the context of STO, personalizing send times to each customer could potentially play a role in influencing CTR positively. By implementing STO strategies the goal is to maximize CTR. The idea behind it is to send emails at the most opportune moment which should increase the chances of capturing the recipients attention and prompting them to take action, resulting in higher CTR and overall positive customer engagement.

3.5.3 Opt-out-rate

Opt-out-rate(OOR), also known as unsubscribe rate is a metric that measures the percentage of recipients that unsubscribe from receiving further emails from the company.

Obtaining the OOR is calculated by dividing the number of recipients who unsubscribe from the received emails by the number of emails sent, then multiplying it by 100 to obtain the percentage.

$$OOR = \frac{\text{Number of people that unsubscribed}}{\text{Number of Delivered Emails}} \cdot 100\% \quad (3.13)$$

In the context of STO, OOR is an important metric for evaluating the connection between the time an email was sent and customer satisfaction. By implementing STO strategies the goal is to minimize OOR.

3.6 Email Types

Every day, a wide array of emails is deployed from IKEA, each created to serve its specific purpose and objective. The type of email selected for use in the construction of the model plays a critical role in how well it predicts both on the basis of live data and evaluation data. Consequently, understanding the nuance behind these emails plays a crucial role in understanding customer interactions and thus improving the likelihood of creating a model that captures the underlying behavioral patterns. The aim of this section is to provide insight into the influence these emails have within the context of STO at IKEA by examining two of the most important email types.

Pull

Pull emails, commonly referred to as triggered emails, are selectively dispatched to customers exclusively in response to specific actions they have taken, rendering them eligible to receive such correspondence. In essence, these emails are triggered by customer-initiated actions, with the primary objective being to attract or "pull" recipients toward engagement. Examples of these triggering events may include customer interactions with IKEA's website or mobile application.

Push

Push emails, commonly known as newsletters, are sent to all subscribers who have consented to receive such communications. Unlike personalized emails, push emails are not tailored to individual recipients and are distributed at a predetermined time set by the department responsible for the send out of newsletters.

Chapter 4

Method

The purpose of this chapter is to describe the methodological framework used to conduct the research. It encompasses an overview of methodological alignments across the models, alongside the design and implementation of the three distinct models: The Historical Interaction-Based Model, The Demographics-Integrated Model, and The Customer Engagement Prediction Model. Each of these models represents a unique strategy aimed at addressing the research problem.

Developing the optimal STO model was a process of iterative development, with the initial two models serving as both benchmarks and foundations for further enhancement. The insights obtained from the first two models shed light on the features and strategies influencing the predictive performance of an STO model. Utilizing these insights, we refined our approach to develop the ultimate iteration: The Customer Engagement Prediction Model.

4.1 Methodological Alignment Across Models

While distinct differences exist among the models, fundamental similarities also emerge. To ensure consistent and comparable results, a uniform approach was adopted across all three models: the utilization of the same email type and the implementation of identical data preprocessing procedures.

4.1.1 Email Type Selection

All three models use the same email type for training and evaluation. Due to the nature of Push and Pull being inherently different, the choice of emails would greatly affect the performance and outcomes of the models. Nevertheless, both of them could offer insight into the behaviors of the customers. Pull emails, characterized by their triggered sending,

offer us a more realistic interaction distribution for each customer, resulting in fewer biases in the data. However, due to the fact that these emails are dependent on customer actions in order to be sent, the volume of data available will be low. In contrast, Push emails provide a greater amount of data for training and evaluation. However, this type of email is sent in bulk, resulting in a spike in customer interaction during the hour the message is sent, as we can see in Figure 4.1 where customers are more likely to interact the first hours after an email is sent, which leads to biases in the data. Overall, the targeted nature of Pull emails may provide a more accurate representation of customer behavior over time, and the low volume was addressed by including a longer observation window.

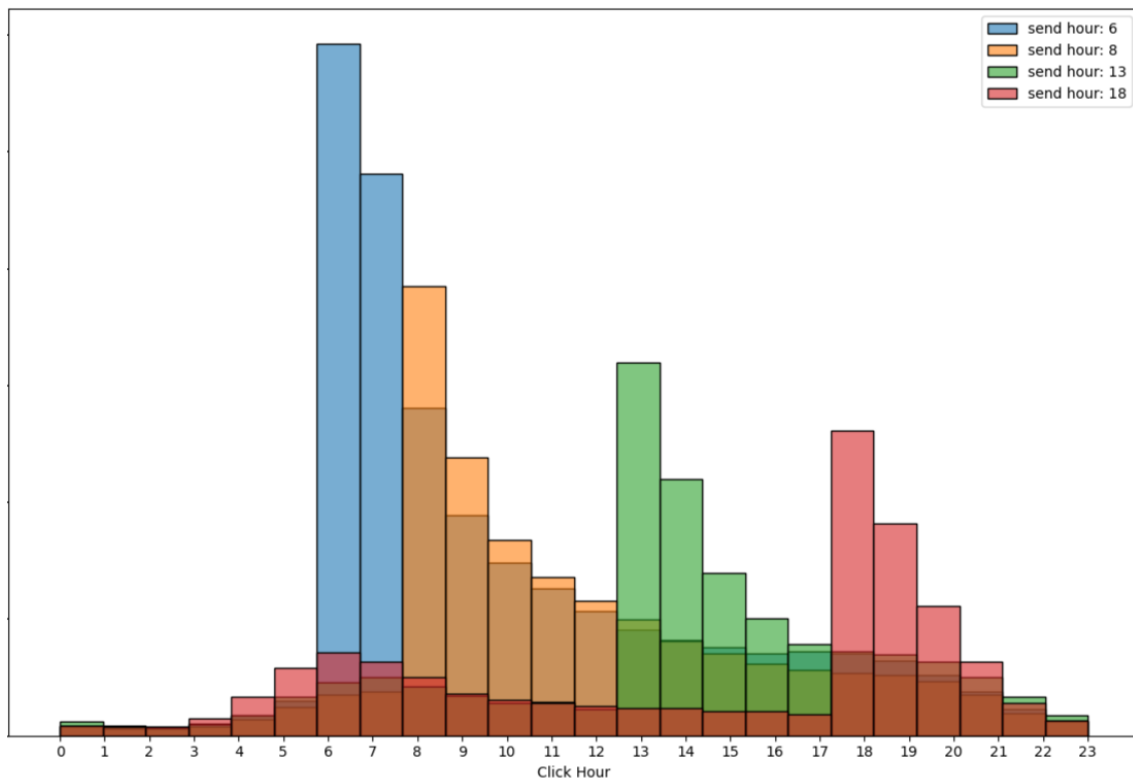


Figure 4.1: Relationship between Send hour and click hour.

4.1.2 Data Preprocessing

Figure 4.2 illustrates the training life cycle for all three models, each of which underwent similar preprocessing procedures. First and foremost, relevant data was collected from IKEA databases. Ensuring data cleanliness was an important part of the preprocessing phase; it essentially involved identifying and dealing with any instances of duplicates or inconsistencies.

Following that, we selected essential features that were significant to each model. We also developed features by altering existing data, resulting in features that were better suited to solving the problem. Finally, we used Pearson’s correlation matrix to examine the relationship between the features and the target, identifying the most important ones. Lastly, the categorical variables were translated into numerical representations. Label encoding was applied to features with an inherent order, while One-Hot encoding was used for unordered or nominal categories.

In order to evaluate the performance of the model, the data was split into an 80/20 split. The features were then trained on various models and fine tuned depending on performance and feature importance. This was done repeatedly until the performance of the model met our expectations.

Training

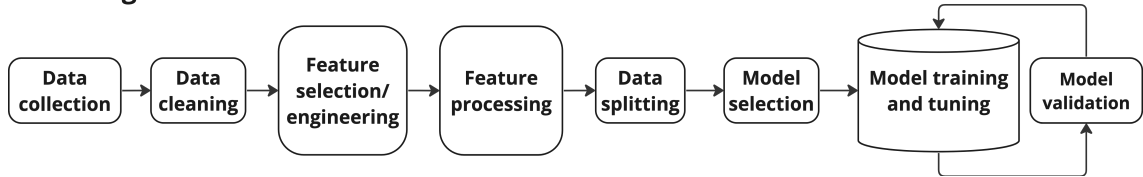


Figure 4.2: Model training life cycle.

4.2 Model 1-Historical Interaction-Based Model

The initial model, also referred to as the base model, was developed solely using historical customer data. The main objective was to gather information on customer interactions, specifically focusing on email clicks and opens within the past 6 months. Predictions were then carried out using a classifier model based on the scheduled weekday for sending emails.

The model's training and testing phases focused on customers who had interacted by clicking on at least two emails during the observed months. Specifically, the model was trained using data from customers who had both clicked on and opened emails within the last 6 months, as illustrated in Figure 4.3. Predictions were then generated for customers who had interacted by clicking on and receiving emails at least once within that same timeframe.

For customers who had no previous click or open activity, the send time was determined using statistics on the most interacted-with hours within their specific market.

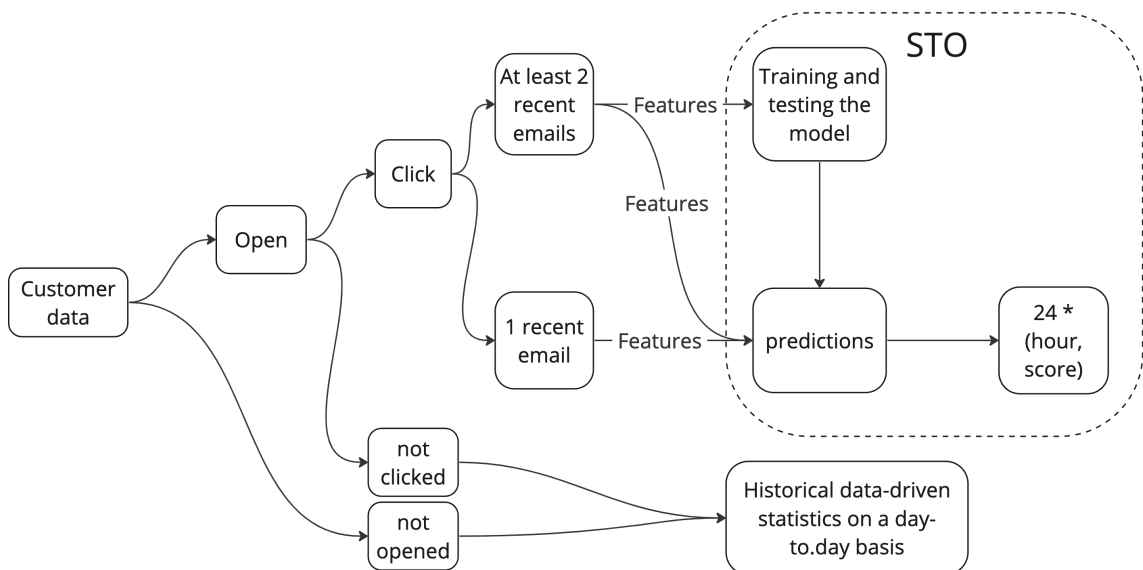


Figure 4.3: Model 1 diagram.

The features used for the model consisted of when the email was sent, when the email was

opened, when the email was interacted with in form of click, time-intervals and interaction-counts.

The base model was divided into two distinct versions, each exploring this approach in different ways.

4.2.1 Model 1.1

In the first version, the training data included the customer's last two interacted emails. The last one was used to extract the dependent variable, and the previous one was used to extract the independent variable, as shown in Figure 4.4. Thus, each customer appeared once in the training set.

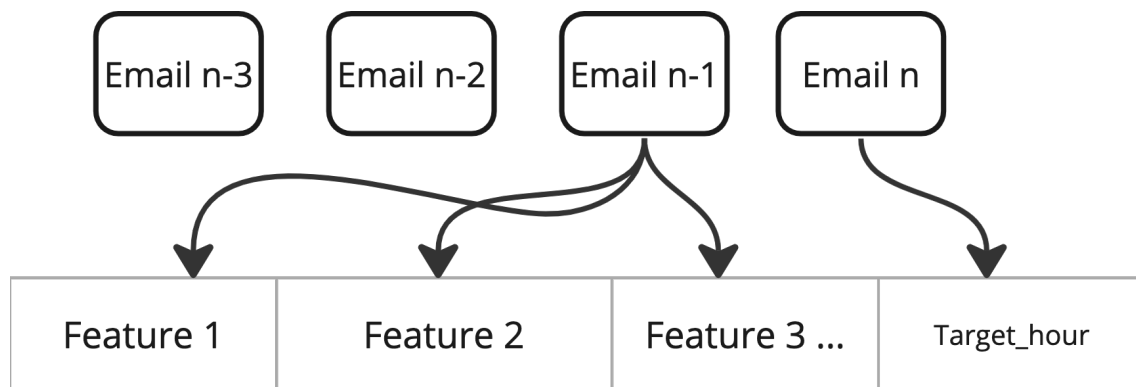


Figure 4.4: Model 1 version 1 data extraction

4.2.2 Model 1.2

In the second version, the independent variables were derived from the subsequent interaction and the prediction was made based on the current interaction. Thus, each customer appears $n-1$ times in the dataset, where n is the number of emails that the customer have interacted with through both click and email within the timeframe as shown in Figure 4.5.

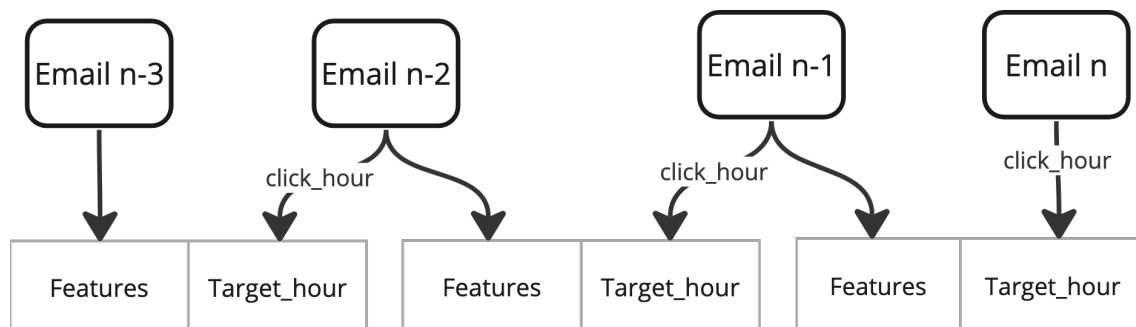


Figure 4.5: Model 1 version 2 data extraction

4.3 Model 2-Demographics-Integrated Model

The idea behind the second model was to incorporate additional customer information with the interaction history of that customer. More specifically, extending the second version of the base model to include demographic data of the customers. A crucial component in the development of this model entailed an examination of the demographic variables influencing customer interactions and the magnitude of associated correlations. This preliminary inquiry was conducted through comprehensive data analysis, employing visualization techniques such as diagrams and graphs, alongside statistical methodologies including Pearson's correlation matrix.

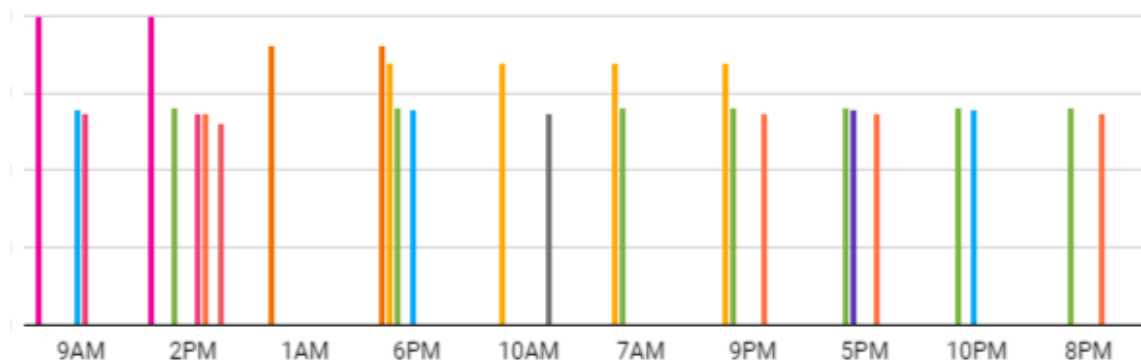


Figure 4.6: Correlation between demographic age groups and customers' interaction time.

In Figure 4.6, we present an illustration depicting the correlation between demographic data and customers' interactions. The analysis reveals distinct interaction preferences among demographic groups; for instance, the "purple" group exhibits a preference for interaction times at either 9 AM or 2 PM, while the "orange" group favors interaction times at 1 AM or 6 PM, and so forth. This examination was systematically conducted for each feature believed to influence the timing of customer interactions with their emails. Features exhibiting weak correlations with customer interactions were consequently excluded from further consideration.

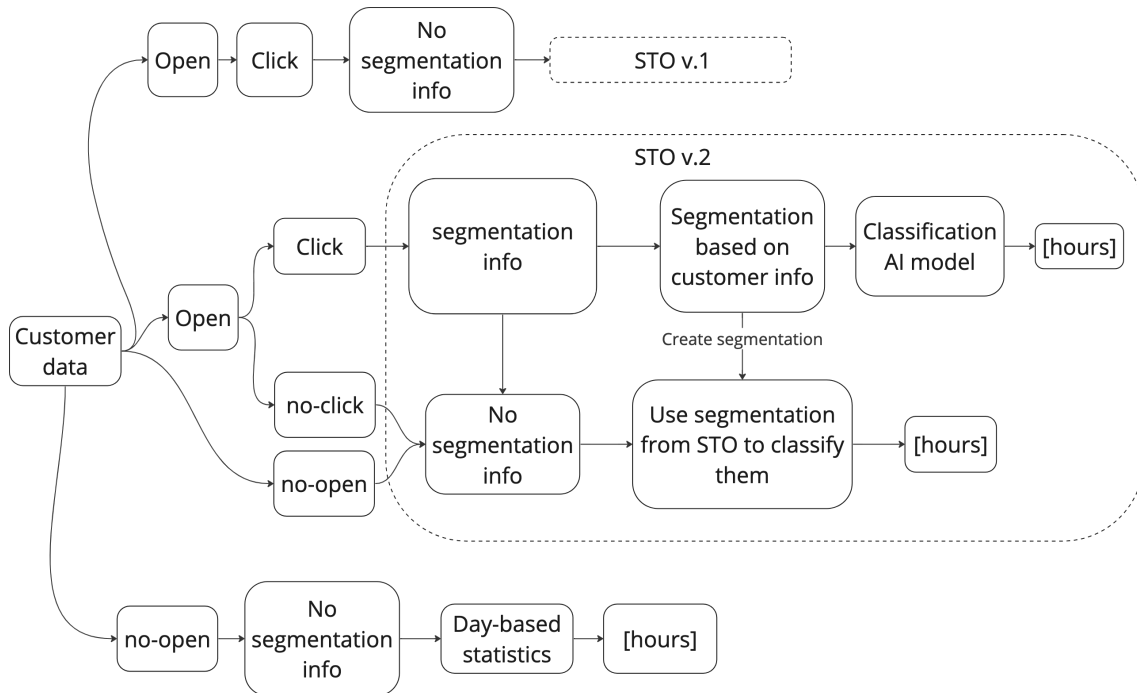


Figure 4.7: Model 2 diagram.

As seen in 4.7 the customer data in model 2 was divided into three section based on the information we had available for that customer. It was divided into the following four sections:

- 1. Customers who have interacted through opens and clicks but have no demographic data:**

Customers who have engaged with at least two different emails by opening and clicking them, but lack demographic data, were included in the initial model. Their sent hour was then predicted using this first model.

- 2. Customers who have interacted through opens and have demographic data:**

For these customers, segmentation was conducted based on their demographic data. A classification model was then applied to this segmented data to predict the optimal time for sending emails.

- 3. Customers who have not interacted but have demographic data:**

For customers who have either not opened any previous email or opened but did not click, and who possess relevant segmentation information, data from the segmentation performed on the previous group—where the relationship between customer segmentation and interaction patterns was established—was utilized to predict the optimal time for this specific group.

- 4. Customers who have not interacted and have no demographic data:**

For the customers that lack any relevant data, the optimal time was derived from day-based statistics. These statistics were based on the customers who have interacted with the emails within the time frame of the training data.

4.4 Model 3-Customer Engagement Prediction Model

The third model is fine-tuned to prioritize the hours when subscribers actively engage, shifting the emphasis from when emails are sent to when subscribers choose to interact. Essentially, this model disregards the timing of historic email dispatch and instead focuses on the hours when subscribers decide to engage, thus the model is scalable to include timestamps of web engagement as an interaction hour in the dataset. The subscribers in the model are segmented based on (i) interaction hours, (ii) recency, frequency, and monetary values from the past year (iii) email domain, and (iv) demographic data.

The model is designed to predict the level of engagement of a subscriber at various times throughout the day. It assigns an engagement score to each subscriber and unique interaction hour, leveraging historical interaction metrics. These historical metrics are analyzed and weighted to reflect the degree of engagement. The weighting process involves grading each subscriber's interaction metrics to emphasize their engagement levels. By doing so, the model aims to provide a nuanced understanding of when subscribers are most likely to engage with marketing emails, which formulates the problem for a regression solution that aims to predict subscribers' level of engagement at a certain hour of the day.

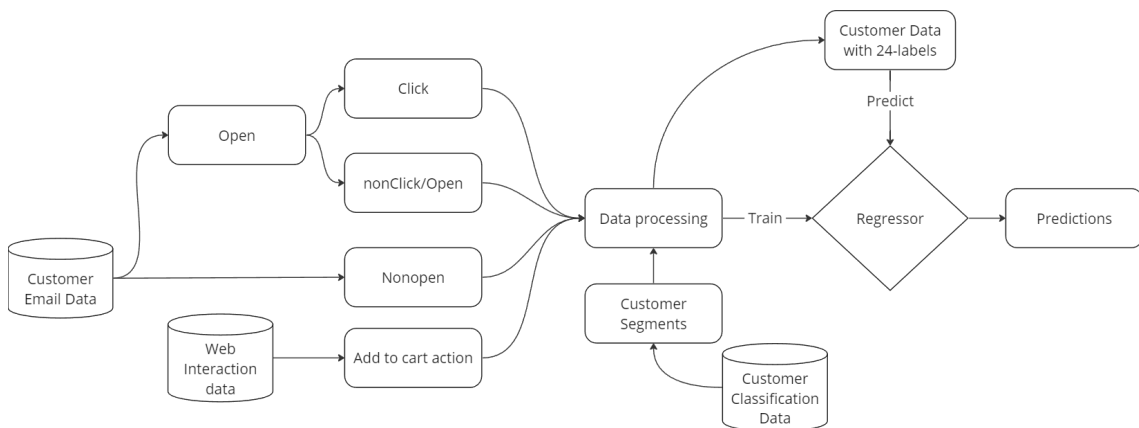


Figure 4.8: Architecture of Model 3.

The strategy of employing a weighted hierarchy system analyzes subscriber interactions based on the time of day. This system assigns distinct weights to three key types of email interactions: clicking, opening, and ignoring as well as one type of web interaction: web action (i.e add to cart) as seen in Figure 4.8. By doing so, the hours of the day that yield higher engagement rates are prioritized, specifically those with higher rates of clicks, and opens as well as web actions, whereas hours, when emails are mostly ignored, are deprioritized. This method involves ranking each historical interaction by hour, using the assigned weights to calculate an overall score for each interaction hour. The process is outlined as follows:

1. **Assign Weights:** Each type of interaction (click, open, web action, ignore) is assigned a specific weight. Clicks, being the most valuable action, receive the highest weight, followed by opens and web action. Ignored emails receive a negative weight or a low

weight to reflect a lack of or low engagement.

2. **Analyze Historical Data:** Historical email interaction data of the past 90 days along with 1 week of web action data are extracted and examined to identify patterns in subscriber behavior, focusing on what times of the day subscribers are most likely to click, open, or ignore emails, and spend time on the web store.
3. **Calculate Scores for Each Hour:** For each customer and historic engagement hour of the day, an engagement score is calculated based on the weighted interactions. This score reflects the subscriber's level of interaction as well as the desirability of sending emails during that hour, with higher scores indicating times when subscribers are more engaged and low or negative scores indicating times when subscribers are low to nonengaged. The formula for this calculation is outlined in the following equation 4.1.

$$\begin{aligned}
 \text{engagement_score} = & \text{click_weight} \cdot \text{normalized_clicks} + \\
 & \text{open_weight} \cdot \text{normalized_opens} + \\
 & \text{ignore_weight} \cdot \text{normalized_ignores} + \\
 & \text{web_action_weight} \cdot \text{normalized_web_actions}
 \end{aligned} \tag{4.1}$$

By implementing this hierarchical system, customer interaction hours are systematically arranged and ranked from best to worst. This ranking serves as the ground truth in the regression model, determining the level of customer engagement during specific hours of the day.

Following the extraction of historical customer interactions, demographic data, RFM (Recency, Frequency, Monetary) values, and email domains, the dataset undergoes preprocessing. Categorical data suitable for one-hot encoding has been transformed accordingly, while non-suitable categorical variables have been label encoded.

The dataset underwent training with various regression models to assess their predictive performance for subsequent A/B testing. The regressors employed included (i) Linear Regression, (ii) Random Forest Regressor, and finally, (iii) XGBOOST Regressor. Through experimentation and tuning, the algorithms were optimized, and the regressor demonstrating the highest R^2 score is selected for the A/B testing phase.

Given the limited data points for numerous customers, particularly lacking interaction data at various hours of the day, new hour labels are generated to fill the gaps as additional data points for each customer. This approach enables the model to predict engagement scores for every customer and hour of the day as visualized in Figure 4.8. The predictions are then ranked based on the engagement scores for the 24 hours of the day. The top 3 hours, determined by the highest predicted engagement scores, are then selected as subscriber predictions for the A/B testing phase.

The model's predictions are stored in Google Cloud Platform (GCP), specifying send-time optimized hours for 20% of the total subscribers in the A/B-testing market. Within this subset, some customers possess sufficient data to be considered statistically significant

predictions, while others with less data have predictions deemed fitting by the model. Customers without stored predictions default to 3 possible fallback hours determined through statistical analysis as send-time optimized hours.

4.4.1 Classification Approach

This section explores the two types of approaches taken to construct and experiment with an STO model using classification.

Binary labels

By reframing the question of how engaged a customer is at a certain hour of the day to whether a customer will engage or not at a certain hour of the day, a classification approach can be employed. This could be done by redefining the labeling strategy. The key task becomes predicting a binary label for customer engagement during specific hours of the day. Two distinct labels are established for this purpose: (i) indicating whether a customer will engage and (ii) not engage during a particular hour.

To implement this classification solution, an essential step involves determining a cutoff value within the engagement scores of the dataset. This threshold acts as a demarcation point, with hours falling below the cutoff labeled as instances of non-engagement and those equal to or above the cutoff as instances of engagement. Subsequently, the model can be trained utilizing well-established classifiers such as XGBOOST, random forest, or logistic regression.

Similar to the regression approach, 24-hour labels are generated for each customer in the prediction phase. This is done to account for hours with no prior historical data to explore if other dispatch hours are more suitable for the customers.

Thereafter, predictions of whether customers will engage or not at different hours of the day can be made and chosen with no regard to ranking. Instead, hours that are predicted to be significant for a subscriber can be randomly extracted with an API solution each time emails are scheduled for dispatch.

24-labels

An alternative classification approach is implemented by restructuring the dataset, moving the interaction hour from the features and using it as a target label. Classifiers inherently calculate probability distributions over the 24 labels and pick the highest probability as the prediction deemed most fitting for the features given. The advantage of this approach is the out of the box inclusion of a ranking system for the hours and eliminating the necessity of generating 24-hour labels for each customer in the prediction phase, unlike previous approaches. For this to be possible, each customer is only allowed one data point in the dataset, otherwise the data will include duplicates of the same subscriber data, creating a prediction conflict. This results in the necessity of filtering a large portion of data by only including the prediction hour that the subscriber has the highest engagement.

4.5 Methodology of A/B Testing the Model

The A/B testing is carried out over a period of two weeks in one market and one IKEA campaign. The model is trained weekly on a 90-day historic data. The email dispatches for a specific campaign is split into 2 versions:

(A) Current model

(B) Thesis model

The volume of the emails is split among the two versions. Each version is assigned to deliver 50% of the overall campaign volume to conduct a fair comparison across the versions.

The evaluation metrics employed for assessment are (i) click-through rates, (ii) open rates, and lastly (iii) opt-out rates. A good model would maximize the overall campaign click-through and open rates and minimize or hold the opt-out rate at a low level.

After the testing period, the metrics produced for each version are compared against each other to determine which model outperforms the other or whether there will be a significant difference in the performance between the models.

Chapter 5

Model Evaluation and Discussion

In this chapter we present the classification results for Model 1 and Model 2. We also present the results of the A/B testing where Model 3 was tested along with the model that IKEA's current model as well as the results for the internal tests of Model 3.

5.1 Model 1-Historical Interaction-Based Mode

The idea behind model 1 was to use a classification model in order to predict exact interaction hour based on previous emails. As we can see in table 5.1 the predictive performance of the model was very low, meaning that the model did not capture the underlying patterns of the data effectively.

Model versions	Accuracy
Version 1	6 %
Version 2	10 %

Table 5.1: Model 1 classification results.

There are numerous reasons for the model's poor performance.

Firstly, the target of the model consisted of 24 labels, representing each individual hour of the day. The problem behind this approach is the granularity of the target, which poses as a challenge since it requires precise predictions at the hour level. Consequently, even a slight deviation from the actual prediction, such as a prediction of 11 instead of 10, would essentially be classified as an incorrect prediction. This fine-grained prediction requirement increases the difficulty of the task and lead to lower overall model performance.

Secondly, the data was not uniformly distributed across the hours of the day, which led to sparse data for certain time intervals. This means that for the hours with fewer training

examples, the model had a hard time learning the patterns effectively. As a result, the model struggled to generalize well to these underrepresented hours, leading to lower accuracy.

Additionally, to be included in the training data, the customers needed at least two interactions with separate emails, and to be able to make predictions for the customers, they needed to have interacted at least one time with an email. Consequently, customers who have only interacted through openings or only interacted once through opens or clicks as well were ignored, despite providing valuable information regarding that customer's interaction. This meant that for the 6 months training data, the base-model could only be applied on 42.8% of the customers within that timeframe. Meaning that the other customers who are not eligible to be used on the model receive a fallback hour.

In the initial iteration, only the two most recent emails with which the customer interacted were considered. This meant that out of, for example, ten emails exchanged during the training data timeframe, only two were processed to be part of the training dataset. This approach presented a significant flaw as it disregarded a substantial portion of the customer's interactions, resulting in the loss of valuable information.

Lastly, the task of predicting the exact hour of customer interaction is inherently a complex problem. The approach of the first models and the complexity of the solution may not have been sufficient to capture the intricacies of the problem. Thus, more sophisticated modeling techniques or feature engineering approaches was necessary to improve performance.

5.2 Model 2-Demographics-Integrated Model

Compared to the base-model, incorporating demographic data as features yielded better results as we can see in Table 5.2.

Demographic data	Accuracy
Demographic feature 1	12.80 %
Demographic feature 2	13.11 %
Demographic feature 3	14.12 %

Table 5.2: Model 1 classification results.

Despite having produced better results, the predictive performance of model 2 was still low. This was largely due to the fact that the previous issues of model 1 had not been addressed correctly. Model 2 still had a target that consisted of 24 labels, and again the fine-grained target approach increased the difficulty of the task.

The unique aspect of model 2 compared to model 1 is that segmentation was used to manage the customers who had no open interactions or had an open interaction but were short of a click interaction. As depicted in Figure 4.6, certain demographic groups displayed preferred preferences regarding interaction times. However, it is noteworthy that not all groups demonstrated such distinct preferences. For instance, the "green" group exhibited a preference across a spectrum of at least 7 different time slots, posing a challenge in establishing a consistent pattern between demographic features and interaction times.

The main difficulty in using demographic data to predict customer behavior is that it's often unclear what exactly influences how a person acts. While many factors might logically

seem important, a major challenge is actually getting access to this information at a company like IKEA.

Despite the poor performances of Model 1 and Model 2, both models added valuable insight into how different features and machine learning approaches affect the predictive performance of a model predicting human behavior. This knowledge was then used in order to build and refine Model 3.

5.3 Model 3-Customer Engagement Prediction Model

Model 3 had the best performance when it came to the internal evaluation and was thus chosen to be part of the A/B testing.

In Table 5.3 we can see the open-rate, click-through-rate and Opt-Out rate. Each of these metrics gives us an insight into how the STO model has performed on live data. The table shows Model 3 without fallback, only the fallback, Model 3 with fallback, and finally the model that IKEA currently use as a benchmark.

Metrics Model	Open-Rate	Click-Through Rate	Opt-Out Rate
IKEA's current model	benchmark	benchmark	benchmark
Model 3 with fallback	0 %	+8 %	+14 %
Model 3, no fallback	+14 %	+22 %	+14 %
Model 3 fallback	-15 %	-6 %	+7 %

Table 5.3: Model 3 A/B testing results.

The performance analysis of Model 3 without fallback reveals its effectiveness on customers present in the training data. Notably, the STO model exhibited superior performance in both open rate and click-through rate metrics, outperforming the fallback mechanism, which relied solely on statistical data and yielded the least favorable outcomes. Consequently, the integration of these two approaches resulted in a diminished overall performance for Model 3 with fallback.

Comparing our model with IKEA's current model, we observe comparable performance in terms of open rate metrics. However, Model 3 marginally outperforms in click-through rate while also showing an increase in the opt-out rate. Although an increased opt-out rate might initially seem negative, it does not necessarily indicate worse performance. Opting out is more often a reflection of content relevance and send frequency rather than timing. Therefore, this can be interpreted as the model sending the message at the most opportune time, prompting the customer to opt out. Moreover, customers opting out of an email campaign list can be advantageous for companies, as it helps refine the audience to those who truly want to receive the emails, thereby increasing overall engagement.

The approach to identifying the optimal algorithm involves training and evaluating the dataset on various algorithms, comparing their performance, and subsequently selecting the algorithm with the highest efficacy for operation in the A/B test. Table 5.4 summarizes the best results of each algorithm after hyper-parameter tuning. The metrics for comparing the

algorithms are primarily R^2 score and MAE along with additional error metrics commonly used for evaluation in regression model performance.

Algorithm \ Metrics	R^2	MAE	MSE	Median-AE	MSLE
Random Forest	0.68	0.2879	0.7028	0.0862	0.0485
Linear Regression	0.38	0.5228	1.3649	0.24	0.1729
XGBOOST	0.74	0.2946	0.8194	0.1034	0.0499

Table 5.4: Model 3 regression results.

As seen in table 5.4, XGBOOST came out on top, attaining an R^2 score of 0.74 and with an MSE of 0.8194. On the other side of the spectrum, is linear regression which achieved the lowest R^2 of 0.38 and MSE of 1.3649 as expected from a linear model that assumes a linear relationship between the features and label rather than a complex nonlinear relationship. Even though Random forest resulted in slightly lower MAE and MSE values, XGBOOST was deemed to be the better algorithm due to attaining higher R^2 score. A lower MAE and MSE indicate that random forest is better at predicting individual data points however they were only marginally lower, and might not explain as much of the variance in the model, hence the lower R^2 in random forest.

After various experiments, tuning, and adjusting parameters, the performance of the model was observed to fluctuate depending on the weights selected to represent the engagement score. Selecting a decimal weight such as 0.7 resulted in a model with lower R^2 than a model trained with an engagement score using integer weights such as 1,2,3. And training the model with no weights at all resulted in the highest performance in regard of R^2 . However, calculating the engagement scores with no weights will add less nuance to the predictions since click rates, open rates, web action rates, and ignore rates are not valued equally. For this reason, we decided to use weights however assigned them integer values rather than decimal values. The algorithms in Table 5.4 were examined with the same weights for a fair comparison.

5.3.1 Time Intervals

Another factor to account for in the performance metrics is the number of time slots. In the proposed model, each customer and unique hour is assigned an engagement score creating a higher variance between the dependent variable e.g. interaction hour, and the independent variable i.e. engagement score. Achieving a higher R^2 score is attainable by grouping hours into larger time slots, for instance, 4-hour slots instead of hourly slots, thereby reducing variance in the data. The rationale behind grouping the hours into larger time slots is grounded in the understanding that, in real-life scenarios, predicting human behavior within broader time frames may be equally or more effective than attempting to predict specific hours. Additionally, one-hour difference in prediction might not have as much of an impact in practice as a 4-hour shift. Rather than predicting top 3 hours for a subscriber which might be adjacent hours in some instances, another approach should be predicting the top 3 time intervals. However, even though this approach might result in a higher R^2 score it would not necessarily indicate a better model in practice. A higher R^2 score in this example highlights that

correlation does not always imply causation. While a high R^2 score is indicative of the model's predictive performance, it may not be accurately tailored to be effective in real-life scenarios. Therefore, experimenting with different time slots should ideally take place in a production environment, where additional metrics come into play and provide a more comprehensive evaluation of the model's practical performance.

5.3.2 Web Data

When adding the web action to the data we were faced with two approaches, (i) use web action timestamps only if the customer has interacted with an email at the same hour in the past 90 days, or (ii) use web actions as an independent data point, regardless if a customer have interacted with an email at the same timestamp or not. For the proposed model we decided to go with the latter approach since it has the potential to add more data for each customer, possibly making the predictions more statistically significant. This approach assumes the premise that subscribers should interact with IKEA emails based on any historical recorded interactions with IKEA e.g. web actions. The first approach instead assumes that subscribers interact with emails at designated hours and web data is to be used as additional pointers to strengthen only the existing relationships in the data concerning subscriber email interactions.

The addition of web actions might come with the risk of adding noise to the data, and excluding it may potentially result in a slightly higher or lower performing R^2 score. However, as mentioned earlier the performance that matters is the one in the production environment, and adding web data as the latter strategy suggests, allows for capturing a broader perspective of subscriber interactions, by spanning across extended hours and a broader customer base, potentially leading to enhanced practical performance.

Including web data is a good way to broaden the customer base and extend the hour coverage in the model data. However, web data introduces two significant challenges, (i) low identification rate and (i) high volume. The low identification implies that a considerable number of customers who interact through the web remain un-logged, making it impossible to identify the users to track historical interactions, unless cookie-based identification is implemented, a practice not supported by IKEA. As for the large volume of web data, the challenge arises from the high cost associated with querying a large dataset through Google Cloud's BigQuery. To address this, we opted to incorporate only one week of web data, focusing solely on one type of web interaction (add-to-cart action). This approach facilitated minimized query costs, while still including web interactions into the model. As for the low identification rate, no modifications were implemented, adhering to IKEA's guidelines.

5.3.3 Predictions

Generating 24 hours, and filling the missing hours of each subscriber before prediction could be perceived as introducing unnecessary noise to the data before making predictions however, the motivation behind adding the full 24 hours of the day for each customer is to improve overall engagement by suggesting new hours to subscribers with only negative or low engagement scores and remaining neutral to subscribers with higher engagement scores. Another strategy to employ would be, generating and adding 24 hours solely for subscribers with low

engagement while keeping the subscribers with a clear interaction pattern (i.e. high engagement score) as they are in the dataset without trying to over-optimize.

An approach that is worth exploring involves leveraging the low engagement scores of subscribers as valuable insight, rather than solely relying on a hierarchical system of 24 hours. By identifying the hours with low or negative engagement scores, these hours are to be blocked for respective subscribers. This approach reduces the likelihood of the model making predictions during hours with low engagements. This type of refinement would be an additional layer of supervising the model to perform more accurate predictions. Furthermore, the prediction dataset will be smaller when blocking hours along with skipping the generation of 24 hours for subscribers with higher engagement, thus cutting the computational costs in Big Query.

Any customer who has not interacted with IKEA in the past 90 days through email nor the past 7 days through the web (i.e. Add to cart action) falls under the fallback category. Customers in the fallback category do not have any customer-specific predicted hours, rather statistical analysis is applied on the predictions of the model, and the top 3 most interacted with hours are utilized as fallback hours for "new" customers. The top 3 fallback hours are ranked to account for different periods of the day depending on when the campaign is triggered for the customer. This approach has the potential problem of creating sending spikes where new customers drive up the interactions the three hours that are used as a fallback. This presents a twofold challenge: Firstly, the concentration of new customer interactions within these three hours introduces biases into the dataset, diminishing the reliability of the model. Secondly, the underlying premise of the STO model is iterative improvement through weekly training, necessitating the exploration of diverse interaction times for each customer. Restricting the options to only three hours limits the model's ability to gain a nuanced understanding of individual customer behaviors across various time slots. An improvement in this approach that could solve the issue is to randomize the top hours for dispatch rather than utilizing a ranking system and thus introduce a better distribution over the hours which has the potential of making the model better over time.

5.3.4 Data Engineering

The current pull email requirements at IKEA stipulate that once an email is triggered for dispatch to a specific customer (pulled), any hour preceding the pull time becomes ineligible for that customer. The email is required to be scheduled the same day as being pulled at any hour post the trigger time. This poses a challenge, as a customer's optimal send-out time may have already passed and therefore the optimal time is no longer eligible since the email must be sent and cannot be scheduled for the next day. The challenge can be addressed either by (i) sending the email right away assuming the next best time is the earliest opportunity, or (ii) using the top 3 predicted hours for each subscriber. The latter approach was implemented in production, however, this resulted in a further challenge that was unsolved in this project due to time limitations. The model sometimes predicted the top 3 hours to be adjacent (e.g. 14, 15, 16). In a scenario where the pull is triggered after the latest predicted hour, the email is sent immediately rather than scheduled by an actual predicted hour of the model. To resolve this, we propose adopting the use of time interval slots rather than hourly slots, thus each subscriber will be assigned top 3 time slots, reducing the risk of deviating from the predictions.

For some customers, the model predictions result in a ranking system where the top hour is at a later time of the day than the hours ranked second and third (e.g. 19, 12, 8). While such customers will most of the time receive the email at the predicted optimal time, the second and third-ranked hours will always be ignored resulting in unnecessary and redundant memory allocation in the database with no practical purpose. This problem arises for the fallback hours as well, if the top-ranked fallback hour is at a later time of the day than the second and third-ranked hours, only the top hours will be of practical use. A way to resolve this problem is to adopt an API solution that randomizes the top hours that are post the trigger time. This allows for more variation in the send-outs, while still maintaining a prediction hour that fits the subscriber.

When constructing the data for model 3, we were faced with two approaches: (i) Train the model on what send hours result in the highest engagement score or (ii) train the model on what engagement hours yield the highest engagement score. The former approach produces a model that predicts the send hour as a successful one even though the customer might interact with the email several hours or days after delivery but ultimately tries to find the send hour/hours that produces the highest engagement. This problem could be resolved by adding weights to the send hours dependent on the send-to-interaction times, assigning send hours with lower send-to-interaction times a larger weight to supervise the model into prioritizing send hours that yield quicker interactions. However, the problem with the first approach arises with incorporating web actions, since web actions do not have send times, and are solely interaction-based. One way to resolve this problem is by applying web action times as supporting weight for send hours. Another issue with targeting send-out times is the occurrence of biases towards one hour more than others. Machine learning is as good as the data used to train the model, and if the email dispatches are more biased towards certain hours of the day, so will the model predictions. The latter approach however strikes directly at the time of engagement and disregards completely when the email was sent. The latter approach felt more natural to go with for incorporating web actions since web actions do not have any send-out times. Further, there is no need to monitor the send-to-interaction times using the latter approach since engagement hours are targeted directly from subscribers' interaction hours. Furthermore, targeting the time of engagements reduces the biases, since the outcome is solely a result of when the subscribers choose to interact. The downside of targeting the interaction times directly arises whether an email should be sent at the time the interaction occurs or prior. A way to determine this is by A/B testing sending on the predicted hour versus scheduling the dispatch one hour before the prediction.

5.3.5 Binary Classification

The binary classification approach was not tested in a production environment, it was however experimented with, to compare against the regression approach. The binary approach has the potential to perform well in a production environment even though the hours are not ranked. Instead, the hours that are predicted to be instances of engagement for each subscriber ought to be stored and linked to the subscriber. When it is time to for dispatch, the hours that are predicted to be instances of engagement can be randomized since there is no way to rank the hours. The advantage of randomizing the hours for each prediction enables more variation in the send times which makes the send-outs less monotonous. The results compiled for the binary approach are presented in the following table:

Algorithm \ Metrics	XGBOOST	Random Forest	Logistic Regression
Precision	0.8934	0.9005	0.8304
Recall	0.8395	0.8011	0.6271
Accuracy	0.9181	0.9017	0.8605
F1 score	0.8657	0.8479	0.7146
Log loss	0.1737	0.2829	0.3618
ROC AUC	0.9763	0.9674	0.8880

Table 5.5: Model 3 binary classification results.

At first glance, the binary classification results look promising, showcasing a superior performance over the regression model, with an F1-score of 0.86 for the XGBOOST Classifier. It's however important to consider the number of labels in the dataset, as models with fewer labels often exhibit better performance than those with multiple labels. The efficacy of the classifier is also heavily influenced by the cut-off point chosen to distinguish engagement instances from non-engagement instances. The cut-off point for the result presented in Table 5.5 is chosen at a value of 0.5.

Even though comparing classification models and regression models head to head is inherently complex, the results of the binary classifier could be interpreted to exhibit superior predictive performance, by simply comparing the R^2 -score to the F1-score. It is crucial to consider the limitations of the binary classifier approach, particularly its inability to rank the predictions. Furthermore, the classifiers pose the risk of predicting all hours to be instances of non-engagement. In such a scenario, the subscriber would either be discarded from the campaign or get assigned one of the fallback hours as a solution.

The binary classification model was ultimately deemed unfitting for IKEA's current ecosystem. The strategy of discarding subscribers from the campaign, as implied by the binary approach, is not aligned with IKEA's requirements nor does the approach provide a ranking system for the prediction hours. The rejection of the approach does however not preclude its applicability in other contexts with different variables.

5.3.6 24-label Classification

The drawback of the 24-hour approach in comparison to the binary, is the filtering of large size of the data since only one record per customer ought to be used i.e. the record with the highest engagement score. This results in a significant loss of valuable data concerning customer behavior. One possibility is to include all the data in the training however, extracting the prediction only from the prediction with the highest engagement score. The results for the 24-hour label approach are the following:

The results demonstrate that the best-performing algorithm is XGBOOST with an F1-score of x, which is a large decrease in performance when compared to a binary classification model rather than a model trained on 24-labels. The F1-score was expected to be lower however, the question of whether a binary model will perform better in production is yet to be tested before any conclusions are drawn. A middle approach to group the labels into time slots and thus reduce the label variance should be explored. The middle approach possibly

boosts the f1-score of the model. Regardless, the 24-label classification did not make it to the production environment either due to the filtering of a large portion of the data.

A variation to the 24-label approach is restructuring the dataset from scratch. Instead of including all interactions of a given hour in one data row (like it is done in model 3), the data is re-engineered to contain one row per interaction in the dataset. This creates a larger data size. However, the time was limited and thus restructuring the dataset to explore this approach was not an option for this project. This should, nevertheless, be considered and compared to the model proposed in this thesis.

5.3.7 A/B Testing

A/B testing has the advantage of giving us insights into how different system works compared to each other. But there are some things to keep in mind when conducting A/B testing. The duration of the experiment is important from the aspect that a test that is run for too long or not long enough might yield results that are not reliable. Due to time constraints, the project could only run for two weeks, which resulted in an insufficient volume of data to generate fully statistically significant results.

Another aspect that could have an impact on the performance of Model 3 are market trends and changes in user behaviors. It would be possible to mitigate this in future work by running the test over multiple time periods and on more markets to account for variability and validate results, which was not possible for this project.

Lastly, in order to make sure there are as little biases in the experiment, we assured random assignment of subjects to each of the three groups, by doing that we minimize sample bias.

For further improvements in the A/B testing, it would be valuable to delve into additional metrics beyond those currently assessed. Specifically, exploring metrics such as revenue gain and purchase rate resulting from the implementation of an STO model could provide deeper insights. These metrics offer a more detailed understanding of the model's impact on the overall effectiveness of email marketing campaigns. By incorporating revenue-related metrics, we can gain a clearer picture of the direct financial benefits derived from optimizing email send times.

5.4 Future Work

Identified opportunities for refinement across model and methodology are listed in the following:

- **Dedicated Database Development:** As an extension to this project, the development of a dedicated database focused on STO presents an intriguing opportunity. Email engagement is influenced by a multitude of factors, and capturing a comprehensive view of user behavior and routines on an individual level is paramount for effective STO implementation. For example, studying a person's morning routine, work hours, commute time, or their preferences regarding email engagement can provide valuable insights into crafting targeted and effective email delivery strategies.

- **Time intervals:** An interesting aspect to further explore is A/B testing hourly slots versus time interval slots and comparing the two approaches with respective yielded click-through, open, and opt-out rates to evaluate which of the two approaches performs more effectively in a production environment.
- **A/B Metrics:** Regarding the metrics of A/B testing, further research should expand the scope of the metrics beyond the conventional parameters of click-through, open, and opt-out rates. The inclusion of conversion rates provides a new dimension to gauge campaign success, measuring not just email interaction metrics but also monetary outcomes. The conversion rates of subscribers could further be incorporated as a method for segmenting customers, in addition to recency, frequency, and monetary values to explore possible performance improvement.
- **Web data:** Future research should incorporate both a larger volume of web data as well as additional types of web actions, which could potentially make a model trained on a dataset of 90 days more statistically significant with its predictions resulting in increased model as well as production performance.
- **Blocking hours:** The model should be supervised further by implementing a strategy of blocking hours for specific customers that exhibit low engagement. Additionally adding missing hours to the data should only be applied to customers with low or negative engagement, to reduce the biases introduced to the data as well as cut query costs.
- **Continuous refinement:** The underlying principle of STO is rooted in the iterative process of continuous refinement enabled by feedback loops and ongoing learning mechanisms. Through careful examination of customer behavior and the systematic tracking of engagement metrics, the model aims to enhance its predictive capabilities by iteratively adjusting send times. This iterative refinement process aims to incrementally improve the model's effectiveness over successive iterations.
- **Extended runtime:** To gather a comprehensive dataset that accurately reflects the nuances of customer engagement patterns, the model requires a longer runtime period. This extended duration allows the model to accumulate a sufficient volume of model-tuned data, which is crucial for robust evaluation. Consequently, a prolonged runtime period is essential to ensure the model's exposure to diverse scenarios and enable it to adapt and optimize its predictions accordingly.
- **Dynamic adjustments:** As a proposed strategy to further improve the performance of the model, we suggest extending the model's runtime duration and allowing it to adjust dynamically based on performance metrics before testing the model out formally. This has the potential of substantially improving the predictive accuracy and overall model effectiveness. Overall, this approach holds the potential to augment the model's ability to capture complex nuances in customer behavior, thereby enhancing its performance across various evaluation metrics.

Chapter 6

Conclusion

This thesis explored the optimization of Send Time Optimization (STO) through a data-driven approach, incorporating web engagements and RFM values. The key findings from our research are promising and address the research questions outlined at the beginning of the study.

First, our exploration of Send Time Optimization yielded promising results in a production environment. The results from the two week long A/B test indicated that the in-house thesis model slightly outperformed the model that IKEA currently uses in terms of both open-rate and click-through rate. This indicates that the incorporation of additional features such as web engagements and RFM values can lead to a more precise understanding of customer behavior, resulting in higher engagement. It's also worth noting that although the opt-out rate increased, this does not necessarily reflect poorer performance. In fact, it could suggest that the message reached customers at a time when they had the opportunity to engage with the email. Since opting out is often more related to content relevance and send frequency than timing alone, this may be more of a reflection of those factors rather than the timing optimization itself.

Second, managing email scheduling for customers with no prior interaction data was addressed by incorporating a fallback mechanism into the model. A significant challenge in the project was that the models were tested on pull emails, meaning that these emails had to be sent to customers regardless of prior engagement. This created the difficulty of predicting optimal send-out times for customers who had no previous interactions with IKEA during the training period. To handle this issue, both Model 2 and Model 3 included a fallback option. By providing a default strategy for scheduling email deliveries to these customers, the fallback ensured that all customers were included in the model. This fallback mechanism was tested in production as part of Model 3 and based on our analysis it is evident that while the fallback strategy is important for ensuring practical application by maximizing coverage, the strategy has certain limitations.

As shown in Table 5.3, Model 3 without fallback had the highest open and click rates, while the fallback on its own produced the lowest results for these metrics. When combining

both approaches, we observed a significant negative impact on the overall performance of the model. This suggests that while the fallback strategy is crucial for maximizing coverage, it requires refinement to better balance effectiveness and inclusivity in future work.

Lastly, the effectiveness of STO strategies was demonstrated through several key factors. The most impactful strategies involved integrating web engagement data and historical email interaction data from customers. While segmenting customers based on factors such as demographics, email domains, and engagement patterns provided a slight improvement, it still enhanced the model's ability to predict optimal send times. Certain demographic groups displayed distinct interaction preferences, and aligning email delivery with these preferences led to better engagement. Though the improvement from demographic segmentation was modest, it contributed to a more targeted and personalized approach, boosting overall engagement.

While these findings are promising, further research is necessary to extend the analysis over a longer period and different customer segments. Additionally, refining the fallback mechanism and exploring other features that could enhance engagement will be essential to fully realize the potential of STO.

References

- [1] Ordinary least squares regression (ols).
<https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols>, 2024. Accessed: 2024-03-10.
- [2] C Araújo, C Soares, I Pereira, D Coelho, M.Â Rebelo, and A Madureira. A novel approach for send time prediction on email marketing. *Applied Sciences*, 12(16), 2022.
- [3] J Brownlee. 4 types of classification tasks in machine learning.
<https://machinelearningmastery.com/types-of-classification-in-machine-learning/>, August 19 2020. Accessed: 2024-03-22.
- [4] T Chen and C Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794. Association for Computing Machinery, 2016.
- [5] Constant Contact. 2023 email marketing statistics: Facts, trends figures.
<https://www.constantcontact.com/blog/email-marketing-research-and-statistics/>, 2023. Accessed: 2024-03-22.
- [6] J Czakon. F1 score vs roc auc vs accuracy vs pr auc: Which evaluation metric should you choose? *neptune*, September 5 2023.
- [7] A Deligiannis, C Argyriou, and D Kourtesis. Predicting the optimal date and time to send personalized marketing messages to repeat buyers. *International Journal of Advanced Computer Science and Applications*, 11(4), 2020.
- [8] M Galster F Quin, D Weynes and C.C Silva. A/b testing: A systemic literature review. *The Journal of Systems Software*, 2024.
- [9] J. Fernando. R-squared: Definition, calculation formula, uses, and limitations.
<https://www.investopedia.com/terms/r/r-squared.asp>, December 13 2023. Accessed: 2024-03-22.

- [10] M. A. Haque, M. A. Rahman, S. S. Al-Bawri, et al. Machine learning-based technique for gain and resonance prediction of mid band 5g yagi antenna. *Scientific Reports*, 13(1):1–22, 2023.
- [11] C. S. W. Hummel. How to use apple’s auto opens safely in your email marketing program. <https://blogs.oracle.com/marketingcloud/post/use-apples-auto-opens-safely-email-marketing>, 2022. Accessed: 2024-03-21.
- [12] Inside Learning Machines. Median absolute error - inside learning machines. https://insidelearningmachines.com/median_absolute_error/, January 26 2024. Accessed: 2024-03-22.
- [13] E. Jackson and R. Agrawal. Performance evaluation of different feature encoding schemes on cybersecurity logs. In *SoutheastCon 2019*, pages 1–9, April 2019.
- [14] X Luo, R Nadanasabapathy, A. Zincir-Heywood, K Gallant, and J Peduruge. Predictive analysis on tracking emails for targeted marketing. In N Japkowicz and S Matwin, editors, *Discovery Science*, pages 116–130, Cham, 2015. Springer International Publishing.
- [15] J Paralic, T Kaszoni, and J Mačina. Predicting suitable time for sending marketing emails. In Jerzy Świątek, Leszek Borzowski, and Zofia Wilimowska, editors, *Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology – ISAT 2019*, pages 189–196, Cham, 2020. Springer International Publishing.
- [16] A Prasad. Regression trees | decision tree for regression | machine learning. <https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning-e4d7525d8047>, February 7 2024. Accessed: 2024-03-10.
- [17] O. Rainio, J. Teuho, and R. Klén. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 2024.
- [18] S Rong and Z Bao-wen. The research of regression model in machine learning field. *MATEC Web Conf. Volume 176*, 2018, 2018.
- [19] P. Sonawane. Xgboost — how does this work. <https://medium.com/@prathameshsonawane/xgboost-how-does-this-work-e1cae7c5b6cb>, December 8 2023. Accessed: 2024-03-10.
- [20] R. Udayakumar, K. Sreekumar, C. S. Nivedha, S. Sivasubramanian, and V. Umadevi. A predictive analysis of autism spectrum disorder using ensemble techniques of machine learning. In *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pages 558–565, November 2023.
- [21] M. Xu, Y. Ao, C. Chang, and Y. Wu. The application of improved k-means clustering algorithm in chinese language education in vocational colleges. In *2023 International Conference on Data Science and Network Security (ICDSNS)*, pages 01–06, July 2023.

- [22] Y Zhang and A Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58(B):308–324, 2015.

Thesis project Data-Driven Strategies for Improving Email Campaign Engagement:

A Send Time Optimization Approach

STUDENTS Dalia Saleh Abbas, Mustafa Al-Jailawi

SUPERVISOR Pontus Giselsson (LTH)

EXAMINER Jacek Malec (LTH)

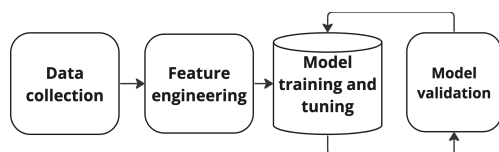
Elevating Email Engagement: The Art and Science of Perfect Send Timing

POPULAR SCIENCE SUMMARY **Dalia Saleh Abbas, Mustafa Al-Jailawi**

If you receive an email in the morning, will that increase the likelihood of you opening and clicking that email? Or maybe in the evening, which one suits you the best? That's what Send Time Optimization (STO) is for, the process of using machine learning to tailor email send times in correspondence with the customers' activity patterns.

The use of email as a medium for exchange of information and promotions has become a major part of communication between companies and customers. In the digital age, where many companies compete to gain the attention of customers, the timing of email delivery has a substantial impact on engagement and response rates. Thus, tailoring send times for emails has become a crucial tactic for maximizing communication efficacy.

In our thesis, we explored what factors influence when a person interacts with their emails. We then utilized these discovered factors to build a Send Time Optimization model. We went through several rounds of model building, by repeating the steps in the diagram below. First, we gathered data. Then, we analyzed which features were most important for understanding how customers interact. Finally, we trained and fine-tuned the model for multiple iterations, analyzing the performance for each iteration and refining the model. This helped us grasp which features and technologies had a positive impact on the model's performance.



The factors found to impact email interactions included: (i) demographic data, (ii) customer loyalty levels, and (iii) interaction data. The model built on these features, named the *Customer Engagement Prediction Model* was then trained and tested on a sample of customers. With high accuracy, it managed to predict when a customer is most likely to interact.

The *Customer Engagement Prediction Model* (CEP) was then trained on a 90 day customer interaction data. It was then tested in production over a period of two weeks and bench-marked against the performance of the model that IKEA currently uses.

The results of the comparison experiment demonstrate a stronger performance by the CEP model compared to the model currently used by IKEA. While both models yielded the same number of email opens, the CEP model exhibits a 22% increase in the number of clicks compared to the current model. In simple terms, it means our model resulted in more customers engaging with IKEA by clicking on email links.

These findings are promising and warrant further investigation over an extended period to enhance their statistical significance.