Using Email Campaign Data to Predict Member Click Behavior

Maria Casas, McKaleb Harlemon, Nathaniel Jones

Kennesaw State University

Submitted on 5/4/2023 to fulfill a requirement for DS 7900

Table of Contents

Executive Summary

This project describes the process and results of developing a *predictive model* to generate the probability that a rewards member will *click* an email from *InterContinental Hotels Group (IHG)*. Ultimately, the predictions or scores can be used to maximize the effectiveness of a marketing email campaign. Four models (XGBoost, Naïve Bayes, Decision Tree, Logistic Regression) were trained and tested to select the best model for the problem's objective:

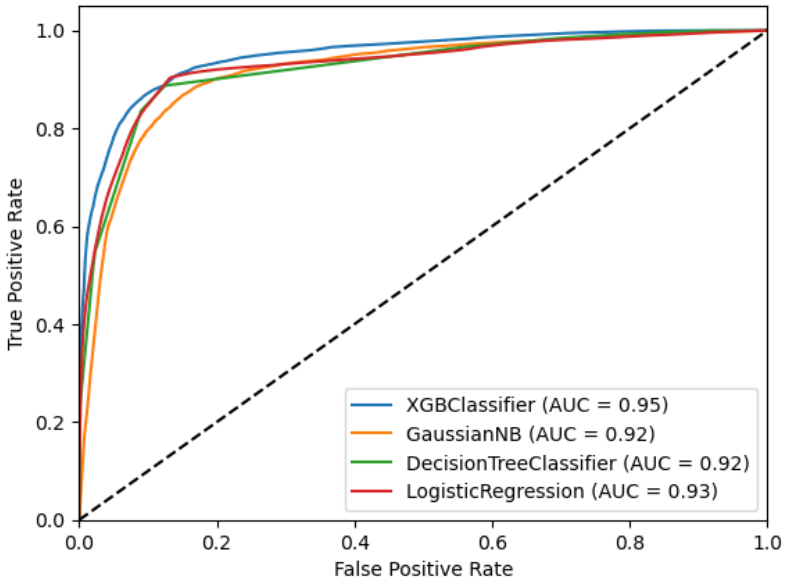
Predict whether a member would click an email

The *main results and recommendations* found by this project are as follows:

1. The *recency* of a member's click associates with a higher probability of a member clicking on another email.
2. An XGBoost model produces the highest Area Under the Curve (AUC: 95 %), lowest proportion of incorrectly identified members who would click (False Negatives: 0.2 %), and the highest proportion of correctly identified members who would click (True Positives: 1.6 %) when predicting whether a member would click an email.
3. If the cost of incorrectly identifying members who would not click is high, then a Naïve Bayes model, compared to the XGBoost model, produces a competitive proportion of True Positives (1.3 %) and False Negatives (0.5 %) while incorrectly identifying 5 % fewer members who would not click.

	Predicted	
Actual	**Click**	**No Click**
	XGBoost	
Click	**1.6%** True Positives	**0.2%** False Negatives
No Click	**12%** False Positives	**86%** True Negatives

	Predicted	
Actual	**Click**	**No Click**
	Naïve Bayes	
Click	**1.3%** True Positives	**0.5%** False Negatives
No Click	**7%** False Positives	**91%** True Negatives



Introduction

Background Information

IHG and Kennesaw State University's Applied Statistics and Analytics department partnered this semester to provide graduate students with a use case for the DS7900 course in Spring semester of 2023. The data provided relates to IHG's commercial email marketing operations, which IHG shared "aims to increase customers' engagement and brand awareness". The ultimate goals of these emails are to direct customers to create bookings, enroll in rewards offers or increase specific brand awareness.

Problem Overview

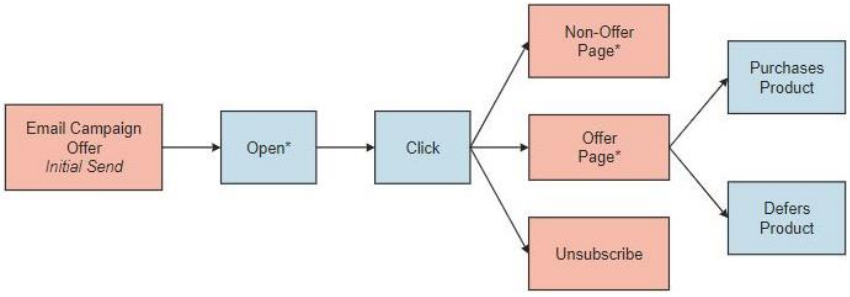
The specific task posed by IHG was to accurately predict whether a member will click an email.

Project Objectives and Significance

The objectives to achieve the goal of accurately predicting include:

- Create a good understanding of the data provided
- Develop analytical plan
- Define parameters such as timeframe to be used for modeling cohort and history
- Feature engineer and feature select influential variables
- Properly merge the datasets into a singular data source for modeling
- Split data into appropriate training and testing datasets
- Build prediction models and train those
- Validate and understand performance of models
- Prepare presentation for IHG to communicate process, results, and recommendations

Leveraging predictive analytics to accurately forecast a customer's likelihood of clicking an email can significantly benefit IHG. By identifying optimal audiences for targeted campaigns, IHG will not only gain a deeper understanding of its member base but also boost the effectiveness of its marketing efforts. Consequently, this data-driven approach will contribute to the company's overarching goal of increasing product purchases and enhancing customer engagement.



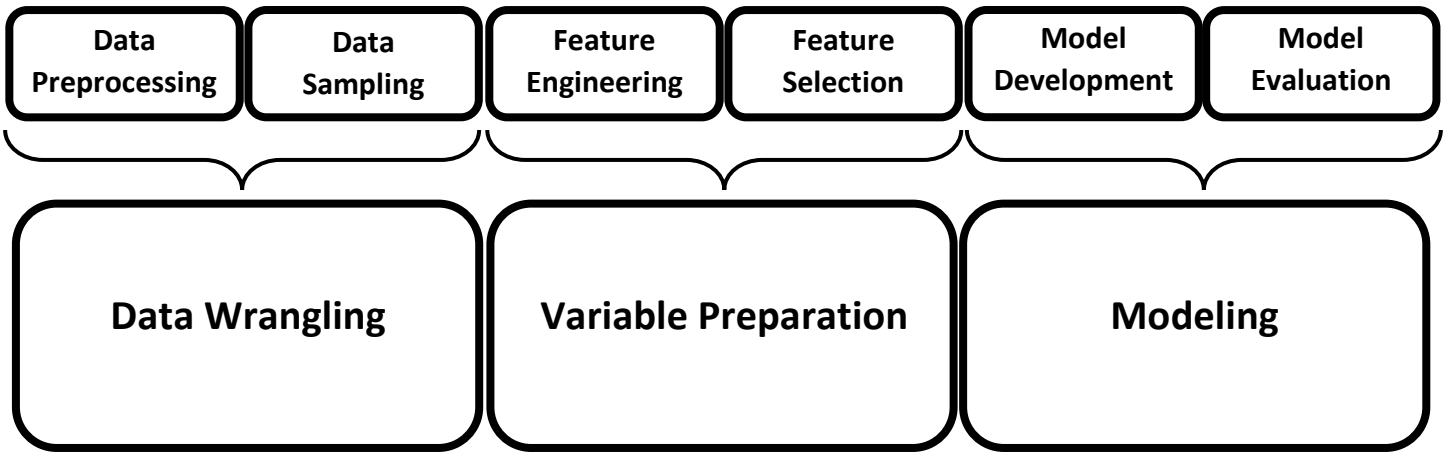
Data Discovery

Data Source

The data for this project was provided by InterContinental Hotels Group (IHG). Three datasets were provided:

- *Email History (192,134,300 observations & 8 variables):*
  - Each observation represents a unique email sent to a member over time. The timespan of emails is between January 1st, 2020, and January 31st, 2022. This file contains the response variable for modeling – 'CLICK', as well as potential predictors of member click behavior. Three variables represent the unit of analysis: member, email campaign, and email send date. (See Table 1)
- *Hotel Travel History (4,724,918 observations & 16 variables):*
  - Each observation represents a unique room consumed by a member over time. This file contains potential predictors of member click behavior. Six variables represent the unit of analysis: member, hotel, hotel stay confirmation, date of stay confirmation, check-in date, and check-out date. (See Table 2)
- *Member Information (1,233,429 observations & 11 variables):*
  - Each observation represents a unique member. This file contains a snapshot of member demographic information as of collection date. (See Table 3)

Each file contains a consistent 'HASH_NBR' that represents a unique member which was used, alongside the time variables, to merge the datasets together. The Email History dataset contains the dependent variable 'CLICK' that will be used as the response variable in the model. Additional features were engineered from the variables in each of the three datasets. The process for the project includes many steps before and after building a model. Each of these processes will be discussed in turn.

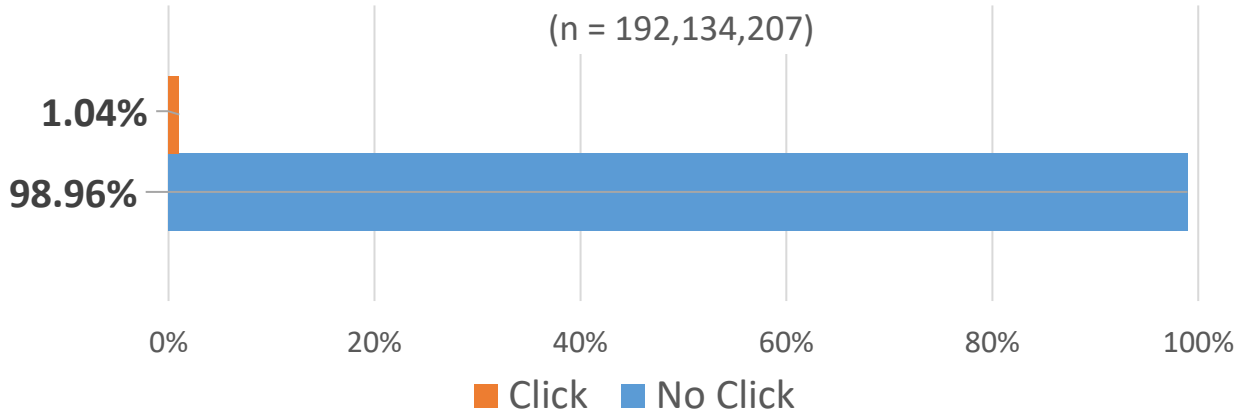


Exploratory Data Analysis

The Response Variable

Within the Email History dataset, the variable 'CLICK' is a binary indicator for whether a member clicks on an email or not. This variable is represented by either a one (1), if the member clicks on the email, or a zero (0), if the member does not click on the email. Figure 1 below displays the volume of 'Clicks' and 'No Clicks' observed in the email history dataset:

FIGURE 1: Volume of Clicks



From this plot, it is observed that click is a rare event and suggests the presence of an imbalanced response variable. This can lead to biased predictions, as the model tends to favor the majority case (members that would not click on an email), potentially resulting in poor performance when identifying instances of when a member would click on an email. Several methods were considered to control for the observed class imbalance bias and will be discussed in the Data Sampling section of this paper.

Email Send Date

The Email History dataset has a date variable ('SEND_DT') that represents the send date of the email to a member. The time span observed in this dataset is between January 1st, 2020, and January 31st, 2022. Each member can be observed receiving multiple emails during this time span that they may or may not have clicked on.

FIGURE 2: Volume of Emails Over Time

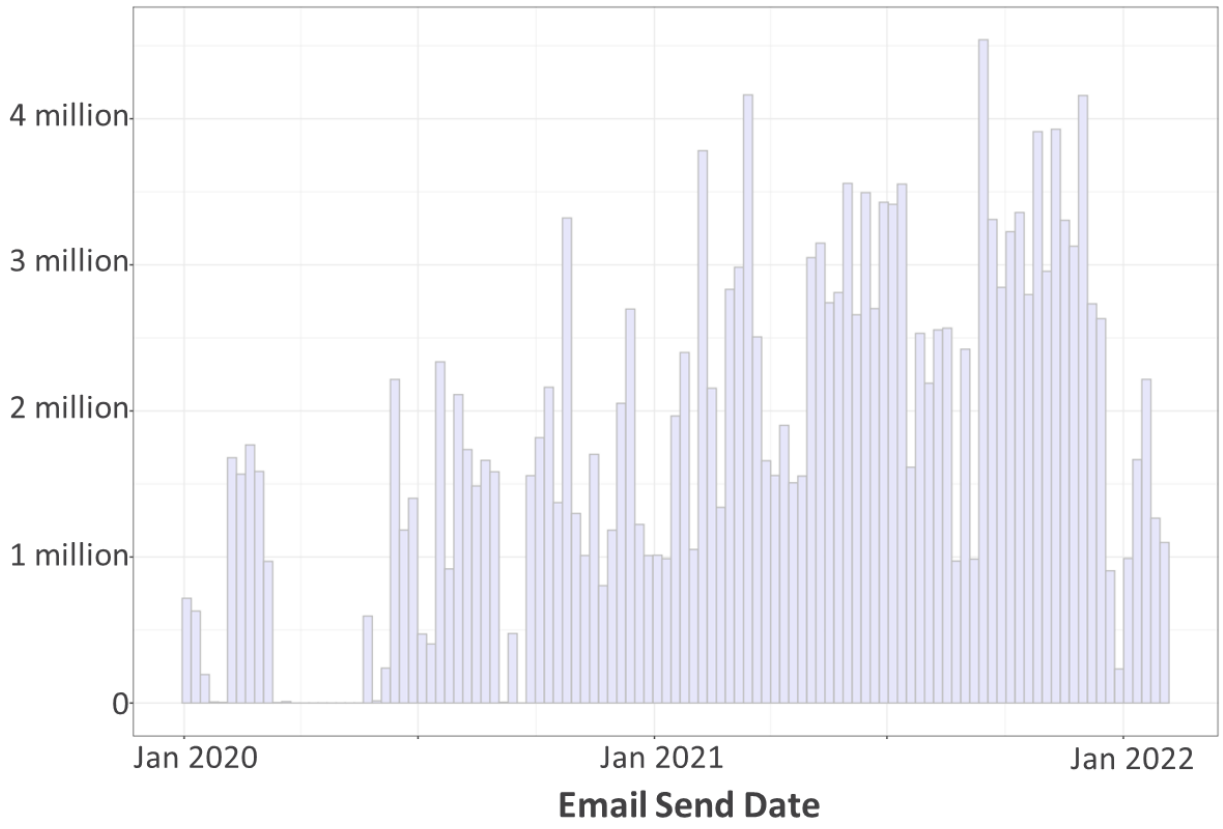


Figure 2 above displays the volume of emails sent to members between January 1st, 2020, and January 31st, 2022. From this plot, the first six months of 2020 display a low frequency in emails sent to members. Continuing along the timespan, an increase in the volume of emails sent can be seen up to the beginning of 2021. This is important because the events of early 2020 may have altered member click behavior.

FIGURE 3: Daily Click-Rate Over Time

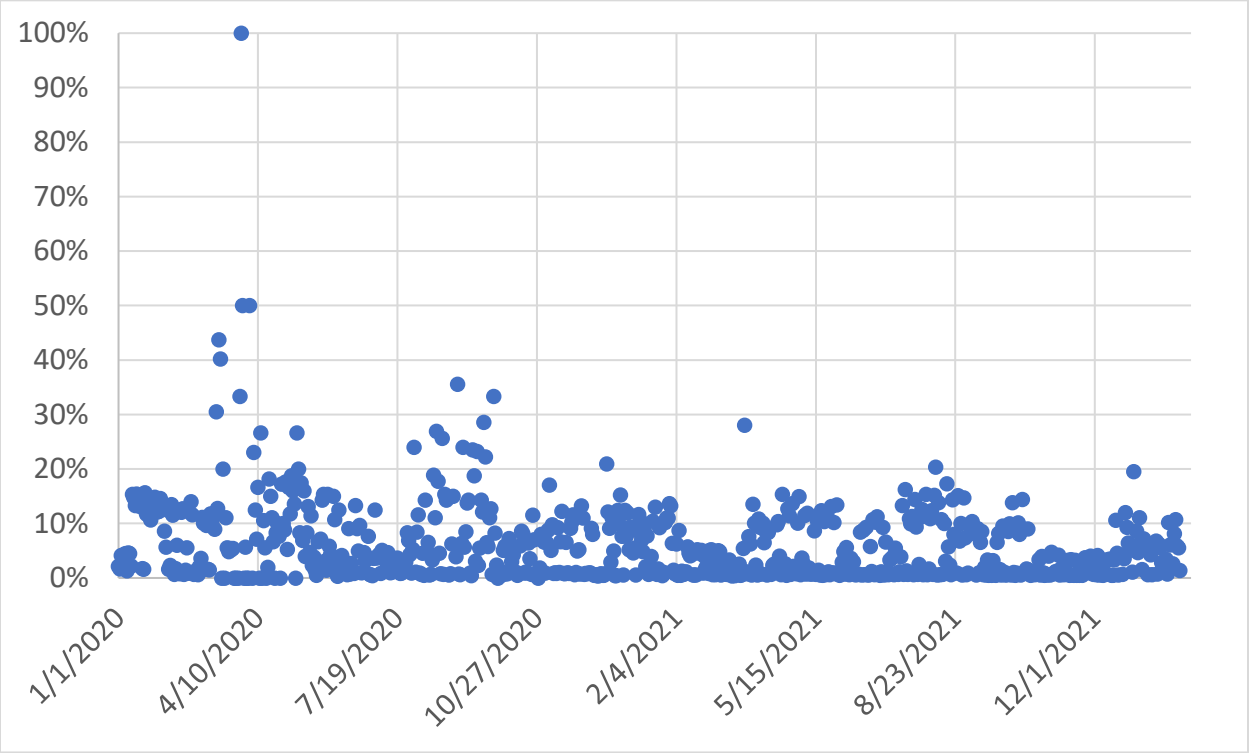


Figure 3 displays the daily click-rate over the observed time span. Although early 2020 was observed to have low frequency of emails sent to members, this graph displays high click rates over the same period. This could suggest that the member click behavior of early 2020 differs from the remaining observed timespan.

Data Wrangling

This phase of the project will wrangle the datasets into a structure good for feature engineering. Due to the time relative information across the Email History and Hotel Travel History datasets, a strategic method to merge these datasets together is needed to ensure data leakage is kept out of the modeling cohort. An improper merge of these datasets may cause future hotel stays to be counted relative to the current email send date of each record. Due to the volume of email records, training a model with all 192 million emails could result in computational resources being unnecessarily used. Randomly reducing the number of emails risks losing valuable information the model needs to learn to predict member click behavior. A sophisticated method of sampling the data is needed to ensure a robust and reliable model.

Data Preprocessing

The first step of this project is to merge the three datasets together in a way that adheres to chronology of time. The Member Information dataset can simply be merged into the Email History by the member ID variable ('HASH_NBR'). The Email History dataset contains emails sent to members over time while the Hotel Travel History contains individual rooms members consume during their travel period. To merge these frames into one, a common join key is needed. To achieve this, the Hotel Travel History was first aggregated to the Hotel Stay level by grouping the data along the member ID ('HASH_NBR') and Hotel Stay Confirmation ID ('CONF_HASH_NBR') variables.

TABLE 4: Example of Stay Aggregation

HASH_NBR	CONF_HASH_NBR	CONF_DT	CK_IN_DT	CK_OUT_DT	HTL_RGN	HTL_CTRY_NM	HTL_CHAIN	GUEST_QTY	REWARD_NT	NBR_OF_NIGHTS	ROOM_REVENUE_USD	BUS_LEIS_IND	HTL_CHAIN_CATEGORY	
001	001	10/20/2021	10/20/2021	10/31/2021	AMER	MEXICO	CHN_3	2	RN_0	11	$121.12	BL_2	CHN_CAT_2	
001	001	10/20/2021	10/20/2021	10/22/2021	AMER	MEXICO	CHN_3	2	RN_0	2	$120.71	BL_2	CHN_CAT_2	
HASH_NBR	CONF_HASH_NBR	CONF_DT	CK_IN_DT	CK_OUT_DT	HTL_RGN	HTL_CTRY_NM	HTL_CHAIN	GUEST_QTY	REWARD_NT	NBR_OF_NIGHTS	ROOM_REVENUE_USD	BUS_LEIS_IND	HTL_CHAIN_CATEGORY	ROOMS
001	001	10/20/2021	10/20/2021	10/31/2021	AMER	MEXICO	CHN_3	4	RN_0	13	$241.83	BL_2	CHN_CAT_2	2

In Table 1, the member ID and stay confirmation ID occurs for each of the room records. To aggregate rooms to stays, the oldest confirmation date and check-in date and the most recent check-out date are retained. Time invariant details of the stay such as hotel country, hotel chain and hotel region remained the same. Variables such as number of nights and room revenue were totaled. Additionally, a new variable called ROOMS was created to summarize the number of rooms included in the stay (original number of room records prior to stay aggregation).

Next, the variables in Hotel Travel History were then regrouped by the member ID and joined into the Email History dataset. Using the hotel check-out date, the listed Hotel Travel History can be partitioned by the email send date to remove hotel stays that happened after the email was sent to the member. The resulting list of variables can be used to engineer features relative to the email send date and prior hotel travel history.

Data Sampling

The Big Data Problem

The full email history includes 192 million emails. Processing this volume of records is computationally expensive. Attempting to work with the full set of observations could potentially result in running out of computational memory or slow processing time during the variable preparation phase of this project. The first action used to reduce the number of emails was to filter out emails sent prior to 2021. Next, emails sent in December of 2021 or January 2022 were partitioned into a validation frame with 7.2 million emails. The remaining 2021 data was split into three-month chunks and a stratified random sample of 4 million email records was taken of each chunk. The resulting modeling cohort contained approximately 18 million emails.

The Class Imbalance Problem

Figure 1 on page 6 of this report displays an imbalance between the frequency of the event class (clicks) and the non-event class (non-clicks). As discussed, this can impact the model and must be handled strategically to ensure the model's reliability. Several methods were considered to control for the observed class imbalance bias:

1. *Over-sampling 'Clicks':* Increases occurrences of rare response class (clicks), balances class distribution, improves model performance, but can lead to overfitting and computational complexity.
2. *Under-sampling 'No Clicks':* Reduces occurrences of common response class (non-clicks), improves model performance, but can result in data loss, misrepresentation of majority class, and overfitting.
3. *SMOTE:* Creates synthetic samples of the minority class (clicks) to alleviate the imbalance while reducing the risk of overfitting but can amplify noise in the minority class and does not account for the underlying distribution of the data.
4. *Stratified k-fold cross-validation:* Divides imbalanced data into k folds while maintaining class distribution, improves accuracy and reduces risk of overfitting, but can be computationally demanding and accuracy depends on chosen value for k and class stratification.

This project used a stratified k-fold cross-validation method to handle the class imbalance. This method ensures that every email in the modeling cohort has a chance to both train and test the model by the end of the modeling process.

Variable Preparation

This phase of the project will wrangle the independent variables into a structure good for modeling. Across the three data frames, a total of 32 variables were given in association with each member. From these, four variables were used to identify unique members, emails, hotels, or hotel stay confirmations and were not used in the feature engineering process. Additionally, four variables were used to identify dates an email or hotel stay was observed. These columns were used to create variables relative to the email send date.

Feature Engineering

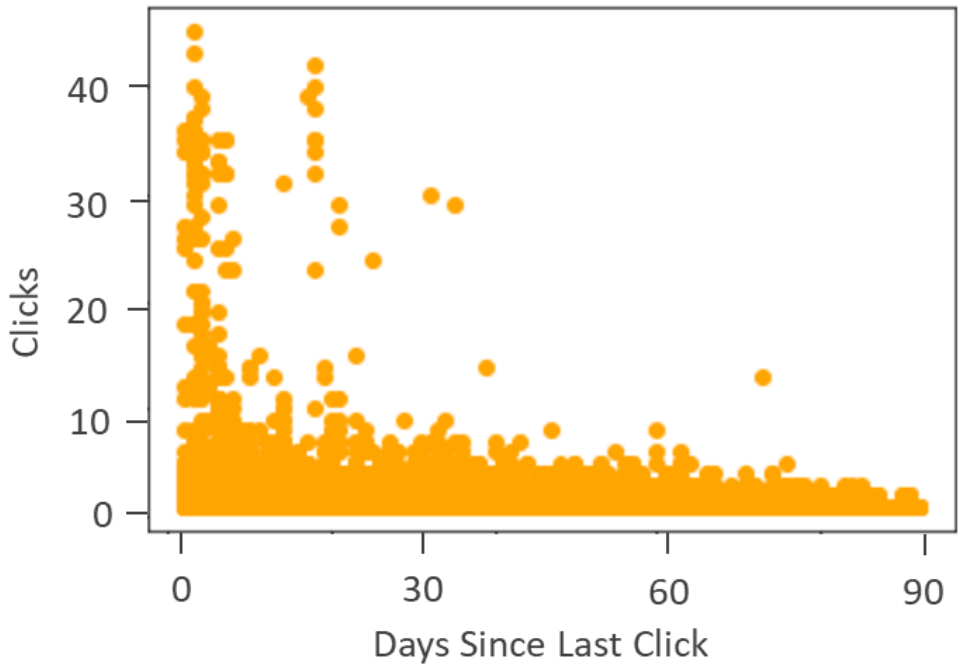
To streamline the process of feature engineering, new features were created from the given variables. Recency, Frequency, and Monetary Value guided the creation and selection process to result in the final set of input variables in the model.

Recency

How recently a member clicked an email link relative to when IHG sent a new email could reasonably give an indication of whether the new email gets the desired response (click). A variable created for this idea was 'Days Since Last Click.' This variable calculated the days between the send date of the new email and the send date of the last email clicked by the member.

Figure 6 shows that high click totals are typically present with lower days since last click. This was an early indicator that recency could be important in predicting a click.

FIGURE 4: Scatterplot Showing Correlation Between Member Clicks and Days Since Last Click

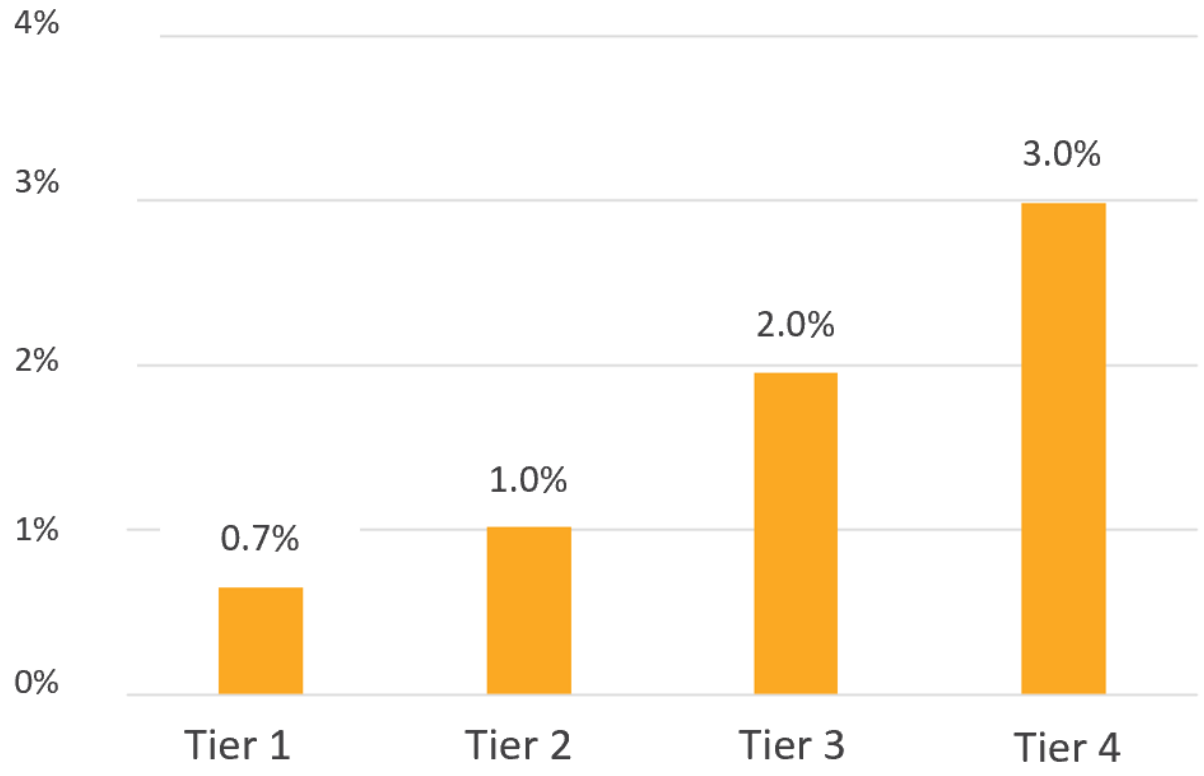


Frequency

In tune with 'recency', how often a member clicks campaign emails is another way to capture rewards members behaviors and habits for the benefit of predicting their interaction with the next email IHG sends. A variable created for frequency was Member Click Rate. This variable represented the proportion of the emails clicked by the member over the total received.

While the overall click-through-rate was found to be roughly 1% for the entire email history dataset, Figure 5 shows that the Member Click Rate increases for each subsequent reward membership tier category. This pattern indicates that frequency may be important in the context of other variables in the model.

FIGURE 5: Click-Rate by Membership Tier

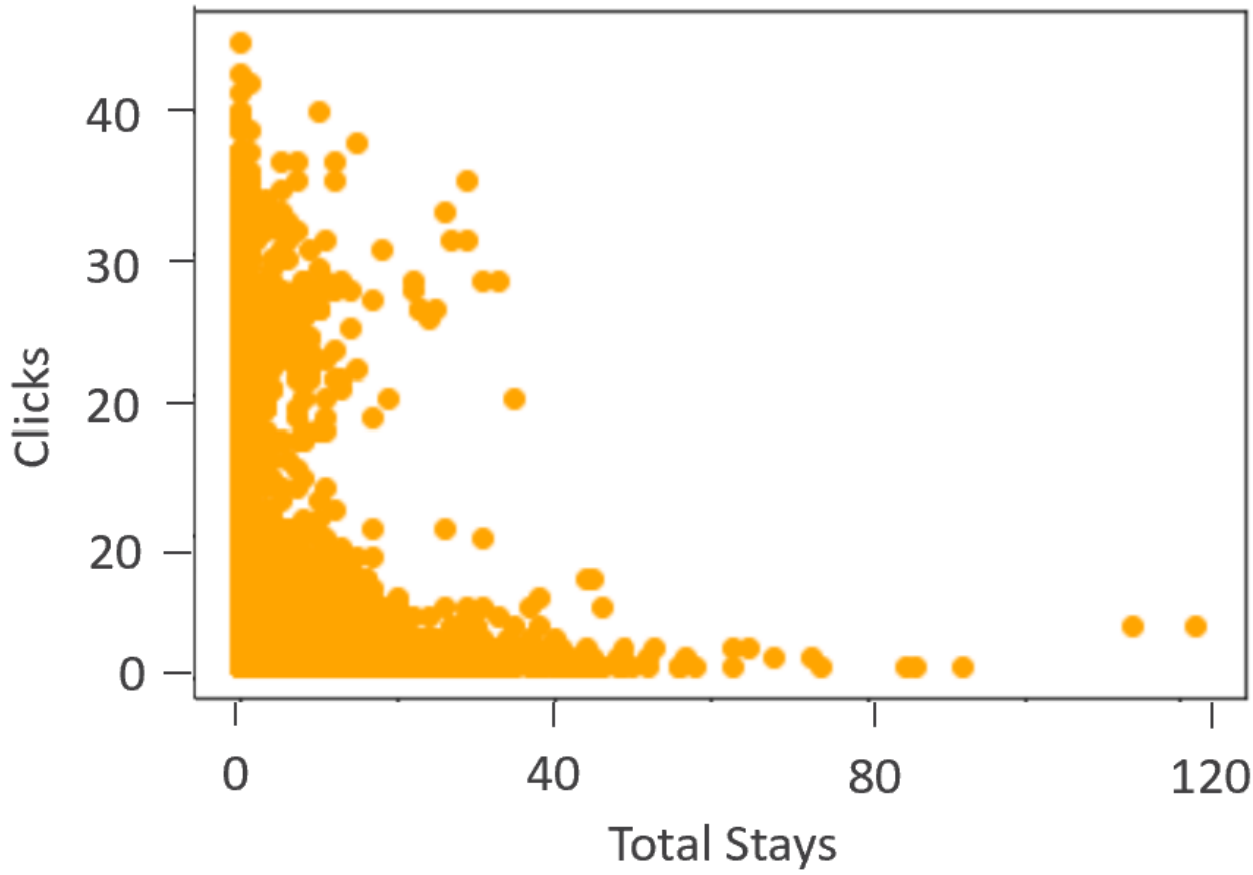


Monetary Value

Through email marketing, the company can increase customers' engagement and awareness of their products. This project considered the time and money a member spends while staying at a hotel potentially important to associate with click. While attempting to capture this relationship several features were engineered and analyzed.

Figure 6 below shows the relationship between clicks and total stays. However, due to the sparse stay data, this relationship could prove not important to the model. From this plot we see that members with a higher number of clicked emails are associated with fewer stays at a hotel.

FIGURE 6: Scatterplot Showing Correlation Between Member Clicks and Total Stays



In total, 298 variables were created from the 32 variables given by the company. Since this project is focusing on an interpretable model, a subset of these 298 variables was selected to maximize parsimony and predictive power. Several methods were considered to reduce the number of dimensions and will be discussed in the next section of this project.

Feature Selection

Selecting features is an important step in the process of creating a reliable and robust model. Including too many features, especially those that are irrelevant or weakly related to the response variable, can lead to overfitting resulting in poor generalization to new, unseen data. In addition, reducing the number of features can lead to more accurate and faster predictions. Fewer features require less computational power and storage, and they can also make it easier for the model to identify the true relationships between the predictors and the response variable. Several methods of feature selection were considered for this project:

1. *Principal Component Analysis:* By transforming the original features into a smaller set of uncorrelated variables, called principal components, that capture most of the variance in the data. The principal components can be ranked by their importance and the least important components can be dropped, resulting in a reduced set of features that still retain most of the information in the original dataset.

2. *Stepwise Selection:* By iteratively adding or removing features based on their individual performance in the model, until the desired level of performance is achieved. This method can be either forward, starting with no features and adding them one at a time, or backward, starting with all features and removing them one at a time. This method runs the risk of selecting an incorrect subset of predictors due to the iterative nature.

3. *Chi-Square Test:* Identifies variables that are highly associated with the response variable. This test can be used to identify variables that are highly associated with the response variable in a dataset by comparing the observed frequencies of the variables with the expected frequencies under the assumption of independence between the variables. A higher Chi-Square suggests the variable has a higher association with the response variable.

This project considered the size of data that could be potentially processed, as well as the infrequency of observed click in selecting variables. In addition, one of the models being considered requires strict independence between the input variables. For these reasons, the Chi-Square test was chosen as the method used in this project.

The Chi-Square Test

For this project, the Chi-Square test was used to identify variables that are highly associated with the response variable. This test can be used to identify variables that are highly associated with the response variable by sorting the test statistics from largest to smallest.

FIGURE 7: The Top-15 Largest Chi-Square Test Statistics

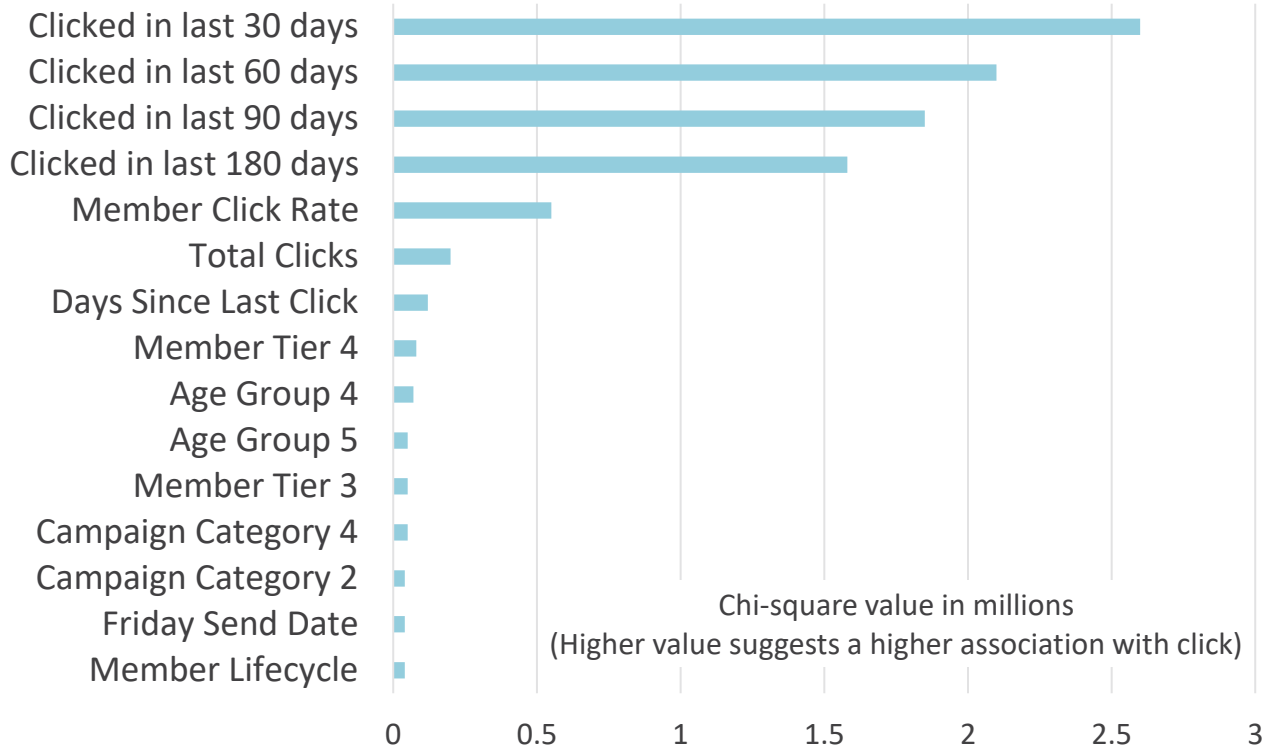


Figure 7 displays the top 15 largest Chi-Square test statistics produced by the engineered variables created for this project. The variable that is most highly associated with the response variable ('Clicked in the last 30 days') is an indicator for whether a member clicked on an email within the past 30 days. Members who have this attribute would have a value of one (1) if they have clicked an email within the past 30 days and a zero (0) for any other case. Further down the plot, the next three variables utilize the same indicator logic, but with a different time window (either 60, 90, or 180 days). Since a member who clicked within the past 30 days would have a 1 for each of these four attributes, using all four attributes in a single model may cause unreliable estimations of the response variable due to the issues of multicollinearity. Therefore, one of these four variables must be selected to represent the other three in the model. In addition, member click rate was created by dividing the total clicks a member has by the number of emails sent to that member. Since member click rate and total clicks are directly related, one must be selected as a representative of the other in the model. To select the best representatives for the model, this project created individual XGBoost classifier (see page 20 for a description of this model structure) models and compared the performance of these models.

The Best Set of Input Variables

The best set of input variables were selected by the model that produced the lowest proportion of False Negative predictions. For similarly performing models, either the lowest proportion of true positive predictions or the lowest proportion of false positive predictions aided in the decision of the better model. The results of this section were used to choose the candidates to build the final predictive model.

The variable 'Days Since Last Click' displayed a high association to the response variable and is not directly related to the other prospective candidates with high association with the response variable. For this reason, this variable was considered for the final set of input variables.

TABLE 5: Member Click Rate vs. Total Clicks

Variable	True Negative	False Positive	False Negative	True Positive	Accuracy
			(n=7,224,150)		
Total Clicks	73.41%	24.83%	0.40%	1.37%	74.77%
Member Click Rate	75.73%	22.50%	0.43%	1.34%	77.07%

The table above displays the performance of an XGBoost Classifier using either a member's total number of clicks or a member's click rate as the input variable. From this table, total clicks produced a slightly lower proportion of false negative predictions (0.40 % vs 0.43 %) and a higher number of true positive predictions (1.37 % vs 1.34 %), but this model produces 1.33 % more false positive predictions. From the results of these models, the number of clicks a member has made should be the chosen variable to represent the other (member click rate), but this variable may potentially bias the model against new members. The number of clicks every member would start with in the observed timespan is zero. As a member's tenure in the observed timespan increases, the number of emails that member has received will also increase. In turn, giving this member a higher potential to have a larger number of total clicks than a member observed less frequently or at a later point in the observed timespan. For example, suppose Member A has received 10 emails and clicked every email and Member B received 1,000 emails but only clicked 10 of them. For either member the value of total clicks is the same (10) but the click rate of Member B (1 %) is significantly different from the click rate of Member A (100 %). Since total clicks could potentially bias the model, member's click rate was chosen as the better representative of the two variables.

TABLE 6: Clicked Within X Days

Variable	True Negative	False Positive	False Negative	True Positive	Accuracy
		(n=7,224,150)			
Clicked in last 180 days	88.77%	9.47%	0.36%	1.41%	90.17%
Clicked in last 90 days	93.09%	5.14%	0.61%	1.15%	94.24%
Clicked in last 60 days	94.53%	3.71%	0.73%	1.03%	95.56%
Clicked in last 30 days	96.40%	1.84%	0.96%	0.80%	97.20%

The table above displays the performance of an XGBoost classifier using the 'clicked within the last X days' variables (where X is either 30, 60, 90, or 180). As discussed in previous sections, these variables are indicators for whether a member has clicked within a window of time (either 30, 60, 90, or 180 days from the email send date) or not. If a member has clicked within 30 days, then this member receives a value of one (1) for each of the four variables. Using all four variables to build the final model could introduce multicollinearity and cause unreliable estimations of the response variable. From the table above, the model using the variable 'clicked in last 180 days' produced the lowest proportion of false negative predictions (0.36 %) and the highest number of true positive predictions (1.41 %), but 9.5 % of predictions made by this model were false positive predictions. In comparison, the model using 'clicked within 30 days' produced 0.5 % less true positive predictions (0.80 % vs 1.41 %) while producing five times fewer false positive predictions. Since the proportion of erroneous predictions made by the model using 'clicked in last 30 days' is significantly less than the model using 'clicked in last 180 days,' the variable 'clicked in last 30 days' was considered the best representative of the model. However, this project was provided unlabeled testing data from February 2022 that the company used as their 'deployment test' of the final model created by this project. Since this data did not include the class label attached to each email, some features engineered related to member clicks, such as 'clicked in last 30 days,' cannot be created for the deployment test data.

For example, suppose a member's most recent email was observed on January 31st, 2022, and that member's most recent click (relative to 1/31/22) is January 2nd, 2022. Since there are 29 days between this member's most recent email send date and the date of their most recent click, this member would have a value of one (1) for 'clicked in the last 30 days.' Now consider an email sent to this member during the 'deployment test' on February 28th, 2022. This email will no longer include this member's most recent click in the 30-day window. For this reason, the variable 'clicked in the last 60 days' was chosen as the best candidate to represent these four variables in the model during the training and validation phase of this project.

To choose the optimal set of variables for the final predictive model, variables directly related to each other and highly associated with 'CLICK' were individually modeled by an XGBoost Classifier and the performance was compared to choose the best representative of the variable group. Through this process the best three variables were chosen for the predictive task:

1. 'Clicked in last 60 days'
2. 'Member Click Rate'
3. 'Days Since Last Click'

Since each of these variables are related to 'CLICK' in some way, a reasonable person may question the correlation between them and if any multicollinearity is present. Figure 8 below shows these variables may be moderately correlated, but since the Variance Inflation Factors where all less than 2 (Table 4) no multicollinearity is present among these variables.

FIGURE 8: Correlation Among the Top Three Variables

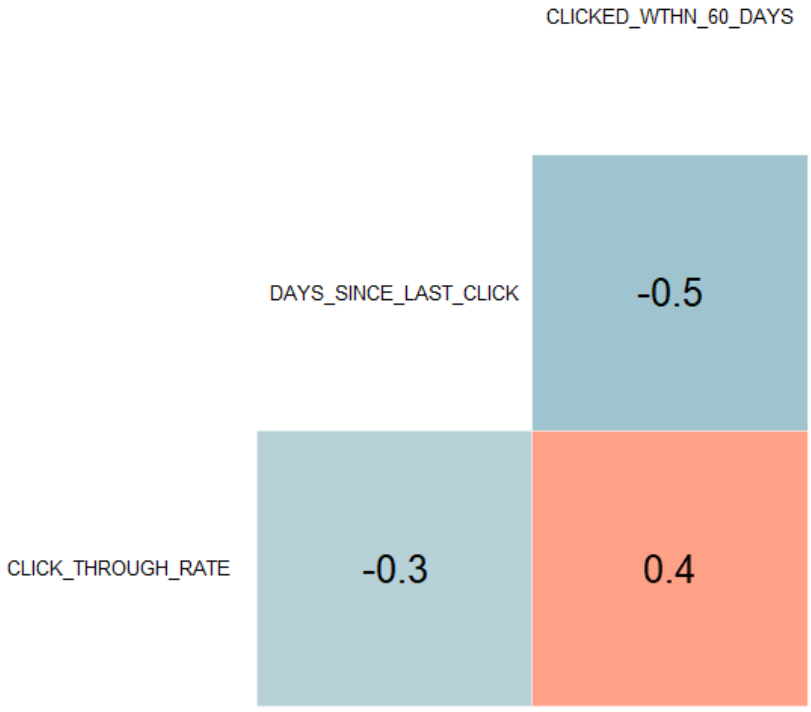


TABLE 7: Variance Inflation Factors (VIF)

Member Click Rate	1.27
Days Since Last Click	1.35
Clicked Within 60 Days	1.48

TABLE 8: Final Selection of Input Variables

Variable List	True Negative	False Positive	False Negative	True Positive	Accuracy
		(n=7,224,150)			
Variable Set 1	85.41%	12.83%	0.18%	1.59%	87.00%
Variable Set 2	85.45%	12.78%	0.14%	1.62%	87.10%

Variable Set 1	• Member Click Rate • Days Since Last Click • Clicked in last 60 days
Variable Set 2	• Member Click Rate • Days Since Last Click • Clicked in last 60 days • Member Demographic Indictors • Email Demographic Indicators

The last step this project took in selecting the final input variable list was to test the performance of two XGBoost classifiers. One model (Variable Set 1) used only the top three variables discussed earlier while the other model (Variable Set 2) uses additional variables (such as the member or email demographic variables) alongside the top three variables.

Variable Set 1 utilized only the best three variables discussed on the previous page. A model using only these three variables performed fairly well compared to the models previously discussed. The prediction made by this model correctly identified 1.6 % of clicks while only incorrectly identified 0.2 %. In addition, the proportion of incorrectly identified members who would not click is 12.8 %.

Personalizing offers to members would allow the company to create a broader marketing campaign that could potentially increase member satisfaction and engagement. This project decided it was important to assess a model that includes demographic indicators (such as age of member, member's gender, member's income category, etc.) to allow for these details to be used in segmenting the member base. A model (Variable Set 2) including the best three variables discussed on the previous page and member/email demographic indicators were created and the performance was compared to Variable Set 1.

This model displayed a slight reduction in the proportion of incorrectly identified members who click (0.14 % vs 0.18 %), an equivalent proportion of correctly identified members who click (1.6 % vs 1.6 %), and incorrectly identified members who would not click (12.8 % vs 12.8 %). Since this model performs equivalently to the model using only the best three variables, the member and email demographic indicators were selected to be in the final set of input variables. The final set of input variables included 57 variables in total. Three were engineered from a member's email history and 54 variables indicate that the member is part of a certain demographic. These variables will be used to create and tune the final models of this project.

Modeling

This phase of the project focuses on creating the model to predict whether a member will click on an email or not from the final set of input variables discussed in the previous section. A plethora of models can be used to achieve a high accuracy, but the differences in modeling structure of some models could potentially capture the variation in member click behavior better than other models. In addition, the original email history contained 192 million records. This project reduced the modeling cohort down to 18 million records through a stratified random sample of emails sent in 2021. Using more data observed in 2021 could potentially create a better model when the computational resources allowed for more data to be processed. For this reason, the scalability and computational efficiency of the model must be considered. Lastly, the event class of the response variable is rarely observed relative to the non-event class. This imbalance can cause the model difficulty in learning the relationship between the independent variables and member click behavior.

Several models were created and tested to select the best modeling structure:

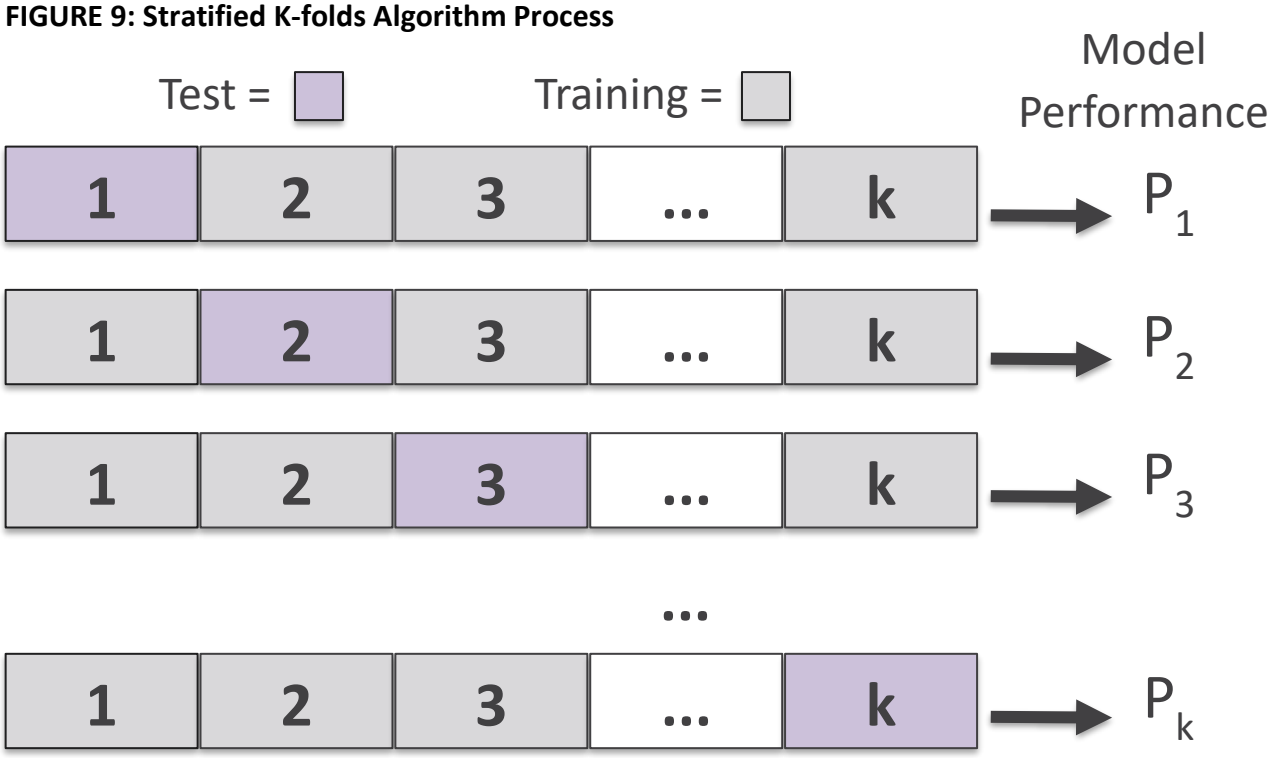
1. *Logistic Regression:* This model is computationally efficient, which is beneficial for large datasets. However, it may struggle with noisy data and may not capture complex relationships or interactions between features.
2. *Naïve Bayes:* This algorithm is fast and can handle high-dimensional datasets effectively. However, it assumes feature independence, which might not be true in practice, and its performance may suffer in the presence of noisy data.
3. *Decision Tree:* Decision trees can capture complex feature interactions and non-linear relationships, but they are prone to overfitting, especially when dealing with noisy data.
4. *Random Forest:* This ensemble method is more robust to noise than single decision trees and reduces overfitting. However, it can be computationally expensive, especially for extremely large datasets.
5. *XGBoost:* XGBoost is efficient in terms of computation and can achieve high predictive accuracy, even with noisy data. However, it may require careful hyperparameter tuning and can be less interpretable than simpler models.

Each of these models were trained on the modeling cohort and evaluated by a variety of metrics such as recall and the F1-score. The method used to train the model was a stratified k-folds cross-validation algorithm and will be discussed in the next section.

Model Training and Validation

As discussed in a prior section, the method used to train the model was a stratified k-folds cross-validation algorithm. This algorithm evaluates the performance of a machine learning model by dividing the data into k equally sized subsets, where each subset has a similar proportion of classes as the overall dataset. In each iteration of the model, a different subset of the data is used to evaluate the performance of the model and after k iteration the overall model performance can be assessed by finding the average performance of each iteration.

FIGURE 9: Stratified K-folds Algorithm Process



$$\frac{P_1 + ... + P_k}{k} = \text{Overall Model Performance}$$

When the response variable is imbalanced, using a higher value for k in stratified k-fold cross-validation can result in a better estimate of the model's performance on the minority class, as each fold is more likely to contain some instances of the minority class. However, it may also increase the risk of overfitting the minority class due to the smaller sample sizes. For this reason, this project used a value of $k = 10$ for the algorithm. After each model was trained through this process the validation dataset (containing emails sent in December 2021 and January 2022) was used as the final evaluation measure of the performance of each model. Specific measures such as the recall or F1-score were used as performance measures, but the business implications must be considered in the analysis of each model's performance.

Model Evaluation

This project focused on maximizing either the recall or F1-score of the model. The recall of a model is a measure of the proportion of actual positive cases (i.e., the member clicks on the email) that are correctly identified by the model. It is a useful metric for evaluating the performance of a model when the cost of missing positive cases is significant. The F1-score of a model is a useful metric for evaluating the overall performance when the response variable is imbalanced. This measure of the model's performance considers both precision (the proportion of predicted positive cases of member click that are correctly identified) and recall (the proportion of actual positive cases of member click that are correctly identified). To fully evaluate the model's performance, the business implications of maximizing either the recall or F1-score must be taken into consideration.

Maximizing the recall of the model will minimize the number of false negative predictions the model makes but will not consider the number of false positives the model may predict. This means a model that predicts that every member would click an email would have a recall of 100 % while the precision of the model is 0 %. If the model instructs the business to email everyone, then why would the business add the extra step of creating a predictive model? Instead, a better performing model must consider the precision while maximizing the recall.

Maximizing the F1-score of the model will balance the number of false negative predictions made by the model with the number of false positive predictions. As a result, more members that are likely to click an email will be predicted not to click. This suggests the opportunity cost of not identifying members who would click is higher, but the risk of spamming members who are not likely to click is low.

Since the company is conducting an email marketing campaign, the risk associated with sending an email to a member that is not likely to click is low. Therefore, this project focused on maximizing the recall while developing the final model. In addition, this project considered the F1-score and the number of false positive and false negative predictions when deciding between similarly performing models. The next section will discuss the specific modeling results found during the creation of this project.

Results

While the models chosen were expected to work well with processed data, the four-performance metrics described in the previous section show which models performed the best in terms of classifying clicks and non-clicks.

TABLE 6: Precision Results by Model

Model	Precision
XGBoost	0.11
Naïve Bayes	0.15
Decision Tree	0.73
Logistic Regression	0.82

The logistic regression model achieved the highest precision rate. This model achieved an 82% for correctly predicted clicks over the total number of predictions.

TABLE 7: Recall Results by Model

Model	Recall
XGBoost	0.89
Naïve Bayes	0.73
Decision Tree	0.24
Logistic Regression	0.14

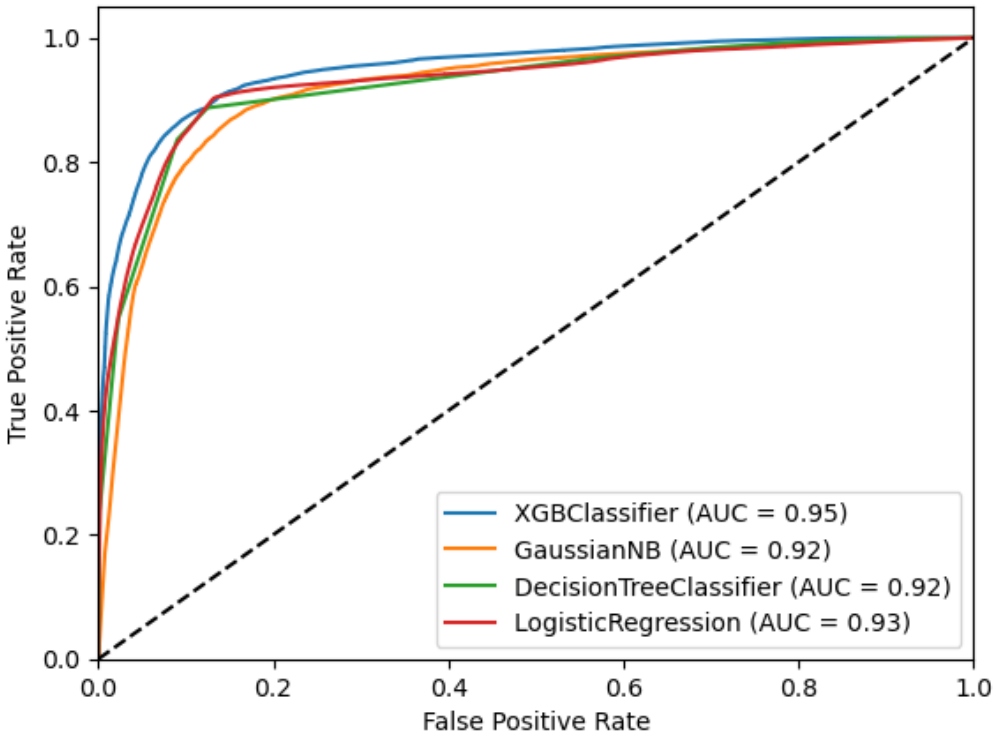
Recall, which measures the actual clicks that are correctly predicted by models, was 89% for the XGBoost. This measure was particularly important due to the value placed on predicting clicks correctly and accurately.

TABLE 8: F1-Score Results by Model

Model	F1-Score
XGBoost	0.20
Naïve Bayes	0.25
Decision Tree	0.36
Logistic Regression	0.24

The F1-score provides a single measure to account for precision and recall. The decision tree achieved an F1-score of 36%, making it the best performing model for this metric.

FIGURE 10: ROC Curve (Area Under Curve) by Model



The area under the curve is a measure of how well models are capable of distinguishing between a click and no click. The XGBoost model outperforms the other models by at least 2 percentage points.

FIGURE 11: Confusion Matrices for Top 3 Models

Actual	Predicted		Predicted		Predicted	
	Click	No Click	Click	No Click	Click	No Click
	XGBoost		Decision Tree		Naïve Bayes	
Click	1.6% True Positives	0.2% False Negatives	0.4% True Positives	1.3% False Negatives	1.3% True Positives	0.5% False Negatives
No Click	12% False Positives	86% True Negatives	0.2% False Positives	98% True Negatives	7% False Positives	91% True Negatives

Figure 11 above shows confusion matrix summaries for the XGBoost, Decision Tree and Naïve Bayes models – the top 3 performing models. True positives, false negatives, false positives, and true negatives are shown as percentages of the total predictions.

All models performed comparatively well. If the goal is to minimize false negatives (predicted not to click, but member did click), XGBoost has the lowest false negative rate and is the optimal model based on that criterion.

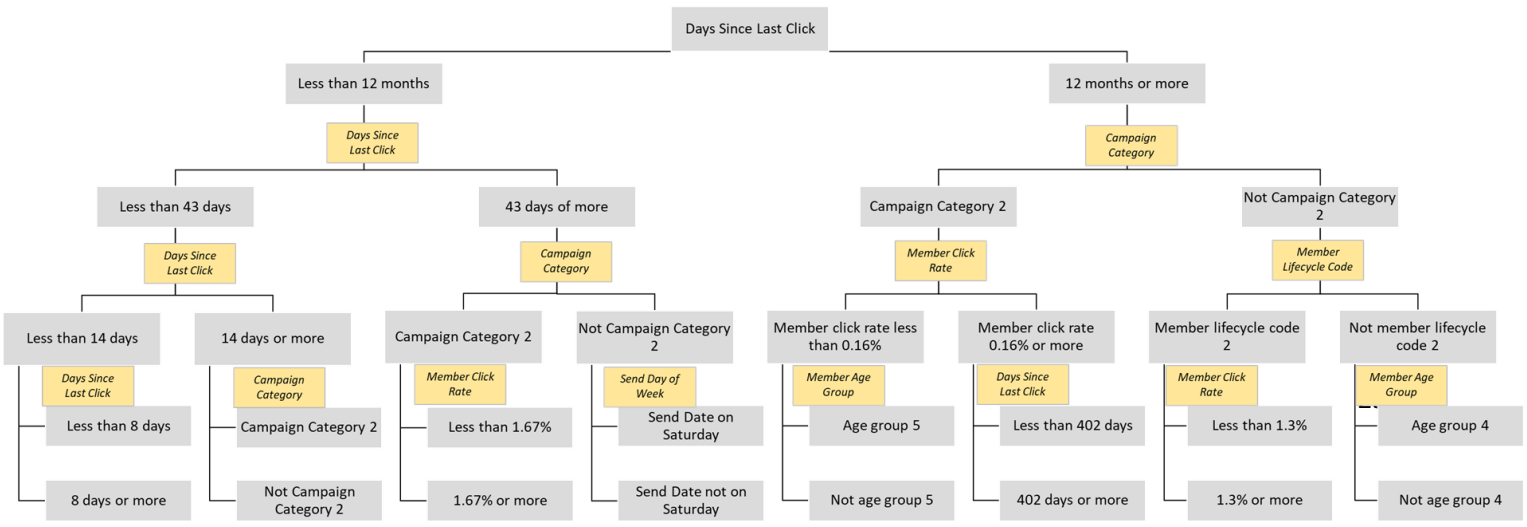
Providing a balance of false predictions, the Decision Tree resulted in 0.2% false positives and 1.3% false negatives (1.5% false predictions compared to the 12.2% from XGBoost and 7.5% from Naïve Bayes).

Still, the Naïve Bayes model best captures the true positives in comparison to the Decision Tree (3x more) and it provides a more balanced false positive to false negative prediction ratio when compared to the XGBoost model.

Assuming minimizing opportunity cost by employing a model that reduces false negatives is the goal for a marketing email campaign, the best model is the XGBoost model shown in Figure 12.

Ultimately, IHG can decide the best model given the parameters and goals for their email campaigns and the level of error in prediction they choose to assume for each outcome.

FIGURE 12: Full Decision Tree Output from XGBoost



Important Variables

The models reviewed had overall good performances in predicting click. The classification or predicting power in the models were driven by the features created as well as those provided originally IHG. While ideas such as recency, frequency and monetary value were the building blocks used for feature engineering, the models were powered by the specific variables shown in Figure 13.

These variables included member and email campaign provided details such as:

- Age Group
- Tier
- Lifecycle Code
- Email Campaign Category
- Email Send Date (where day of week, month and other time data was extracted)
- Income Group (only identified by XGBoost)
- Enrollment Channel (only identified by XGBoost)
- Gender (only identified by Naïve Bayes)

While the rest of the important variables were featured and represented the recency and frequency categories:

- Days Since Last Click
- Member Click Rate
- Clicks in Last 60 Days

FIGURE 13: Variables of Importance

Identified by XGBoost Only	Important Variables Identified by Both Models	Identified by Naive Bayes Only
• Member Income Group • Member Enroll Channel	• Days Since Last Click • Member Click Rate • Member Age Group • Member Lifecycle • Member Tier • Campaign Category • Email Send Date (ex: day of week, month)	• Clicked in Last 60 Days • Member Gender

Conclusion

Like with any task of this caliber, understanding the goals and data set the project up for success. When working with large, disjoined data and ultimately attempting to provide actionable and interpretable insights, starting with learning, planning, and asking questions, can make the task clearer, if not easier.

Much time was spent with the feature engineering and selection phase because a prediction model can only capture the needed information to predict accurately when there is a robust set of features. Understanding what works and is used in marketing was a good step in that direction. From that, the overarching ideas of recency, frequency and monetary value came about.

Carefully engineering the variables while thoughtfully considering the history to be used for emails received and stay history paved the way for variables that captured different time frames for the members' behaviors and interactions. This phase resulted in nearly 300 total variables from which to choose during the selection process.

Choosing the four models to use included researching the best methods to work with the specifications of the data such as the large number of categorical variables, the class imbalance, and the missing data found in the stay history. The models chosen are proven to work well with these structures and resulted in good performances in predicting whether a member would click an email.

Given the performance metrics, the XGBoost model gives reliable and interpretable results that can be used to identify target audiences with the best relative returns on investment.

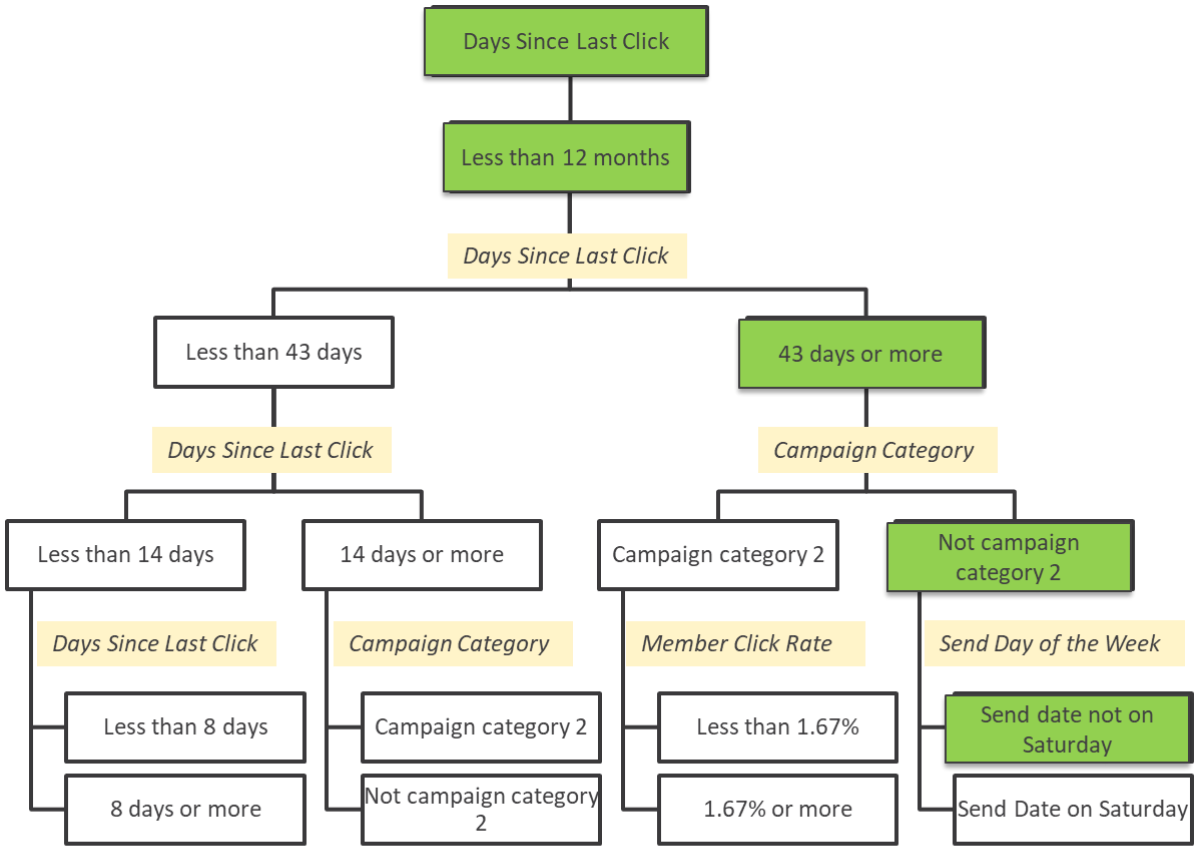
A couple of key findings:

- The recency of a member's click associates with a higher probability of a member clicking on another email.
- An XGBoost model produces the highest Area Under the Curve (AUC: 95 %), lowest proportion of incorrectly identified members who would click (False Negatives: 0.2 %), and the highest proportion of correctly identified members who would click (True Positives: 1.6 %) .

Furthermore, because the final outputs of the XGBoost model are decision trees, those can be further analyzed to extract insights as illustrated in Figure 14.

By following one of the decision tree branches, it was uncovered that members who (1) clicked within the last year, (2) clicked more than 43 days ago, (3) received the email on a day other than Saturday, and (4) received an email from a campaign category other than Campaign Category 2 make up **6%** of the emails sent in the validation set used to understand the model performance (January 2022), yet they yielded **28%** of all the clicks in that same time frame.

FIGURE 14: Insights from XGBoost Tree



In conclusion, this paper effectively demonstrates the value of using data-driven approaches in email marketing campaigns to identify and target audiences with the highest probability of engagement. By leveraging advanced machine learning techniques and carefully engineered features, businesses can make better-informed decisions to optimize their marketing efforts and maximize the return on investment.

Appendix

I. Data Dictionary

Table 1: Email History Data Dictionary

Name	Type	Definition
HASH_NBR	ID	Unique Member ID
CAMPAIGN_NBR	ID	Unique Campaign Code
CLICK	Binary	Indicator if Member clicked
SEND_DT	Datetime	Date Campaign/Promotion Sent
MBR_TIER	String	Member Tier as of send date
MBR_PRGM_ACTV	String	Member Lifecycle code as of send date
CAMPAIGN_NM	String	Campaign Category
UNSUB_IND	Binary	Indicator if Member unsubscribed

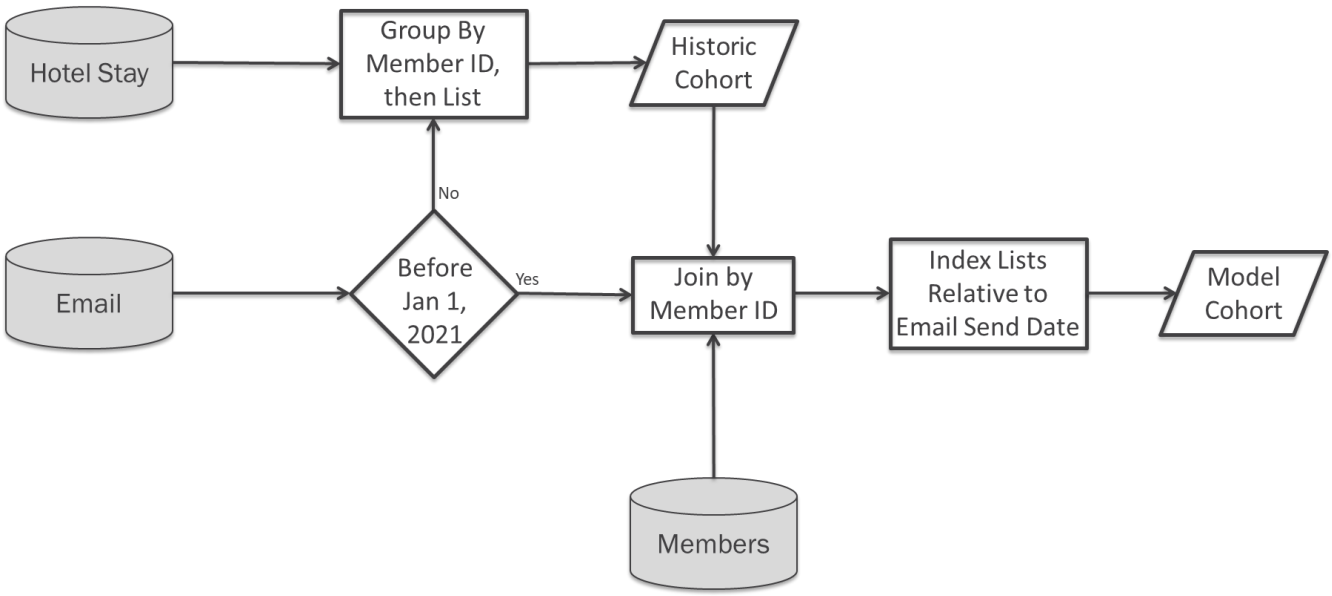
Table 2: Hotel Travel History Data Dictionary

Name	Type	Definition
HASH_NBR	ID	Unique Member ID
CONF_HASH_NBR	ID	Confirmation Code
CONF_DT	datetime	Initial date of booking
CK_IN_DT	datetime	Hotel Check In Date
CK_OUT_DT	datetime	Hotel Check Out Date
HTL_HASH_NBR	ID	Unique Hotel Code
HTL_RGN	string	Hotel Region as of stay date
HTL_CTRY_NM	string	Hotel Country as of stay date
HTL_CITY_NM	string	Hotel City as of stay date
HTL_CHAIN	string	Hotel Brand/Chain
HTL_CHAIN_CTGRY	string	Hotel Brand/Chain Category - e.g. Mainstream
BUS_LEIS_IND	string	Stay Purpose Ind
REWARD_NT	string	Reward Night Category
GUEST_QTY	integer	Number of guests in stay
NBR_OF_NIGHTS	integer	Total Room Nights consumed

Table 3: Member Information Data Dictionary

Name	Type	Definition
HASH_NBR	ID	Unique Member ID
ENROLL_DT	datetime	Member Enrollment Date
ENROLL_CHANNEL	string	Member Enrollment Channel
MBR_REGION	string	Region
MBR_SUBREGION	string	Subregion
STATE_NM	string	Country
CITY_NM	string	State
AGE_CD	string	Age as of Jan 2022
INCOME_CD	string	Income as of Jan 2022
GENDER_CD	string	Gender

II. Data Merge Diagram



III. Data Grouping Example

- Slice of a Member's History

Member ID	Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code
-4369011546959550000	6838726584746060000	2022-01-26	5 days	No Click	0	CC_4	TIER_4	LFC_3
-4369011546959550000	-6617856287572160000	2022-01-22	9 days	No Click	0	CC_2	TIER_4	LFC_3
-4369011546959550000	-8445748874843500000	2022-01-21	10 days	Click	0	CC_1	TIER_4	LFC_3
-4369011546959550000	-1217983810879220000	2022-01-18	13 days	No Click	0	CC_4	TIER_4	LFC_3
-4369011546959550000	1471910069918070000	2022-01-18	13 days	No Click	0	CC_4	TIER_4	LFC_3
-4369011546959550000	-5557044721485620000	2020-05-21	620 days	No Click	0	CC_3	TIER_4	LFC_3
-4369011546959550000	-161241378635773000	2020-03-06	696 days	No Click	0	CC_1	TIER_4	LFC_3
-4369011546959550000	4161280516338980000	2020-02-04	727 days	Click	0	CC_1	TIER_4	LFC_3
-4369011546959550000	6680421968641630000	2020-01-19	743 days	No Click	0	CC_2	TIER_4	LFC_3
-4369011546959550000	6085139861998290000	2020-01-02	760 days	No Click	0	CC_1	TIER_4	LFC_3

- Grouped by Member ID

	Variable Lists after Grouping by Member ID							
Member ID	Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code
-4369011546959550000	[6838726584746060000, -6617856287572160000, -8445748874843500000, -1217983810879220000, 1471910069918070000, -5557044721485620000, -161241378635773000, 4161280516338980000, 6680421968641630000, 6085139861998290000]	[2022-01-26, 2022-01-22, 2022-01-21, 2022-01-18, 2022-01-18, 2020-05-21, 2020-03-06, 2020-02-04, 2020-01-19, 2020-01-02]	[5 days, 9 days, 10 days, 13 days, 13 days, 620 days, 696 days, 727 days, 743 days, 760 days]	[No Click, No Click, Click, No Click, No Click, No Click, No Click, Click, No Click, No Click]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[CC_4, CC_2, CC_1, CC_4, CC_4, CC_3, CC_1, CC_1, CC_2, CC_1]	[TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4]	[LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3]

IV. Data Grouping Example (Training Data Snippet)

Join Lists Back into Model Cohort

Member ID	Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code	Variable Lists after Grouping by Member ID							
									Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code
-4369011546959550000	6838726584746060000	2022-01-26	5 days	No Click	0	CC_4	TIER_4	LFC_3	[6838726584746060000, -6617856287572160000, -8445748874843500000, -1217983810879220000, 1471910069918070000, -5557044721485620000, -161241378635773000, 4161280516338980000, 6680421968641630000, 6085139861998290000]	[2022-01-26, 2022-01-22, 2022-01-21, 2022-01-18, 2022-01-18, 2020-05-21, 2020-03-06, 2020-02-04, 2020-01-19, 2020-01-02]	[5 days, 9 days, 10 days, 13 days, 13 days, 620 days, 696 days, 727 days, 743 days, 760 days]	[No Click, No Click, Click, No Click, No Click, No Click, No Click, Click, No Click, No Click]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[CC_4, CC_2, CC_1, CC_4, CC_4, CC_3, CC_1, CC_1, CC_2, CC_1]	[TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4, TIER_4]	[LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3, LFC_3]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Email Send Date	List of Email Send Date
2022-01-26	[2022-01-26, 2022-01-22, 2022-01-21, 2022-01-18, 2022-01-18, 2020-05-21, 2020-03-06, 2020-02-04, 2020-01-19, 2020-01-02]

List Index of Emails Sent Prior to Email Send Date
[0
1,
2,
3,
4,
5,
6,
7,
8,
9]

Full Index List

Variable Lists after Grouping by Member ID									
Email Send Date	List Index of Emails Sent Prior to Email Send Date	Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code
2022-01-26	[0,	[6838726584746060000,	[2022-01-26,	[5 days,	[No Click,	[0,	[CC_4,	[TIER_4,	[LFC_3,
	1,	-6617856287572160000,	2022-01-22,	9 days,	No Click,	0,	CC_2,	TIER_4,	LFC_3,
	2,	-8445748874843500000,	2022-01-21,	10 days,	Click,	0,	CC_1,	TIER_4,	LFC_3,
	3,	-1217983810879220000,	2022-01-18,	13 days,	No Click,	0,	CC_4,	TIER_4,	LFC_3,
	4,	1471910069918070000,	2022-01-18,	13 days,	No Click,	0,	CC_4,	TIER_4,	LFC_3,
	5,	-5557044721485620000,	2020-05-21,	620 days,	No Click,	0,	CC_3,	TIER_4,	LFC_3,
	6,	-161241378635773000,	2020-03-06,	696 days,	No Click,	0,	CC_1,	TIER_4,	LFC_3,
	7,	4161280516338980000,	2020-02-04,	727 days,	Click,	0,	CC_1,	TIER_4,	LFC_3,
	8,	6680421968641630000,	2020-01-19,	743 days,	No Click,	0,	CC_2,	TIER_4,	LFC_3,
	9]	6085139861998290000]	2020-01-02]	760 days]	No Click]	0]	CC_1]	TIER_4]	LFC_3]

Index List After Removal

Variable Lists after Grouping by Member ID									
Email Send Date	List Index of Emails Sent Prior to Email Send Date	Campaign ID	Email Send Date	Number of Days behind (1/31/22)	Click	Unsubscribe Indicator	Campaign Category	Member Tier	Lifecycle Code
2022-01-26	[	[	[	[	[	[	[	[	[
	1,	-6617856287572160000,	2022-01-22,	9 days,	No Click,	0,	CC_2,	TIER_4,	LFC_3,
	2,	-8445748874843500000,	2022-01-21,	10 days,	Click,	0,	CC_1,	TIER_4,	LFC_3,
	3,	-1217983810879220000,	2022-01-18,	13 days,	No Click,	0,	CC_4,	TIER_4,	LFC_3,
	4,	1471910069918070000,	2022-01-18,	13 days,	No Click,	0,	CC_4,	TIER_4,	LFC_3,
	5,	-5557044721485620000,	2020-05-21,	620 days,	No Click,	0,	CC_3,	TIER_4,	LFC_3,
	6,	-161241378635773000,	2020-03-06,	696 days,	No Click,	0,	CC_1,	TIER_4,	LFC_3,
	7,	4161280516338980000,	2020-02-04,	727 days,	Click,	0,	CC_1,	TIER_4,	LFC_3,
	8,	6680421968641630000,	2020-01-19,	743 days,	No Click,	0,	CC_2,	TIER_4,	LFC_3,
	9]	6085139861998290000]	2020-01-02]	760 days]	No Click]	0]	CC_1]	TIER_4]	LFC_3]

V. XGBoost Decision Tree Example II