

CAPSTONE PROJECT REPORT:

STARBUCKS PROJECT

DEFINITION

PROJECT OVERVIEW

Starbucks, one of the largest global coffeehouse chains, relies heavily on customer engagement strategies to maintain its competitive advantage. A key aspect of their marketing involves the use of personalized promotional offers ([Jacinta Dias, 2024](#)) to attract and retain customers. These offers—such as discounts and Buy-One-Get-One (BOGO) deals—are typically delivered through the Starbucks Rewards program and digital channels. However, predicting whether a customer will complete an offer remains a challenge due to variations in customer demographics and offer types.

Personalized marketing has become increasingly relevant in the past decade, as companies recognize the value of tailoring offers to specific customer segments. ([Yixuan Zhu, 2024](#)) For Starbucks, understanding how customer attributes such as age, income, and gender influence the likelihood of completing an offer can optimize resource allocation and improve campaign effectiveness. Previous research has highlighted the importance of customer segmentation and behavioral analysis in predicting responses to promotions, yet there is still a gap in accurately modeling these interactions in real-world scenarios.

This project aims to build a machine learning model that predicts the probability of a customer completing a Starbucks offer based on demographic profiles and offer characteristics. By addressing this problem, the model can provide actionable insights to enhance personalization, increase offer completion rates, and ultimately drive customer satisfaction and loyalty.

The project involves working with three datasets, all in JSON format. The first dataset is named “portfolio.json” and contains details about different types of offers. Each record in this file provides information on the channels through which the offers were delivered, the difficulty

(amount required to spend) to qualify for the offer, the duration of each offer, the offer ID, the type of offer, and the associated reward.

The second dataset, “profile.json,” contains demographic information for individual customers. The attributes recorded for each customer include their age, the date they became a member, gender, customer ID, and income.

The third dataset, “transcript.json,” records all transactions and interactions with offers from the beginning of data collection. Each record in the transcript dataset has four main features: event, customer ID, time, and value. The **event** attribute indicates whether an offer was received, viewed, or completed, or whether a normal transaction took place. The **customer ID** associates the transaction with a specific customer. The **time** feature represents the number of hours since the start of data collection when the transaction occurred. The **value** attribute contains the offer ID if the event is related to an offer.

These datasets are not real customer data; instead, they are simulations generated by Starbucks to mimic actual customer behavior. The datasets will be wrangled and combined to create a unified dataset. During preprocessing, informational offers (which do not prompt responses) and transactions unrelated to offers will be excluded. Only offers that were viewed by customers will be considered for modeling. The unified dataset, containing details of offer types and customer demographics, will then be used to train a machine learning model to make predictions.

PROBLEM STATEMENT

The primary problem is determining which factors influence whether a customer will complete a Starbucks promotional offer. Despite the abundance of customer demographic data and detailed information on the nature of the offer, it isn’t always clear which combinations lead to higher offer completion rates. This confusion results in inefficient marketing strategies and missed opportunities for optimizing customer engagement. A potential solution is to build an ML model that predicts the likelihood of offer completion based on customer demographics and

characteristics. The success of the solution can be measured using an AREA UNDER THE CURVE score.

The entire end-to-end solution to the problem will be solved using AWS Sagemaker. Sagemaker Studio notebooks will be instantiated and used to preprocess the data and send to S3 buckets. To solve the problem of predicting whether a customer will complete a Starbucks promotional offer, a machine learning classification model, such as XGBoost, will be developed. This model will be created using a Sagemaker training job. The model will use customer demographic data (e.g., age, income, gender) and offer characteristics (e.g., offer type, duration) as input features. The model's performance will be evaluated using AUC to measure its effectiveness in distinguishing between customers who complete offers and those who don't. By accurately predicting offer completion, the solution can help optimize marketing strategies and improve customer engagement.

METRICS

The primary evaluation metric used in this project is the Area Under the ROC Curve (AUC), which measures the model's ability to distinguish between customers who complete offers and those who do not. AUC is chosen because it is robust to class imbalance and provides a comprehensive measure of model performance. The benchmark model is expected to have an AUC of 0.5, while the solution model must exceed this to demonstrate effectiveness.

ANALYSIS

DATA EXPLORATION AND PREPROCESSING

The initial rows of each dataset were examined to gain a general understanding of the structure and contents. Additionally, the "info" method was employed to identify any potential missing data. In the "profile" dataset, which provides demographic information on all customers, missing data was observed in the "gender" and "income" columns. While every other column contains 17,000

entries, these two columns have only 14,825 entries, indicating that the missing values might be attributed to incomplete demographic information for certain customers. Conversely, the remaining two datasets were found to be complete, with no missing values. A bar plot, as shown below, illustrates the distribution of each event in the transcript column.

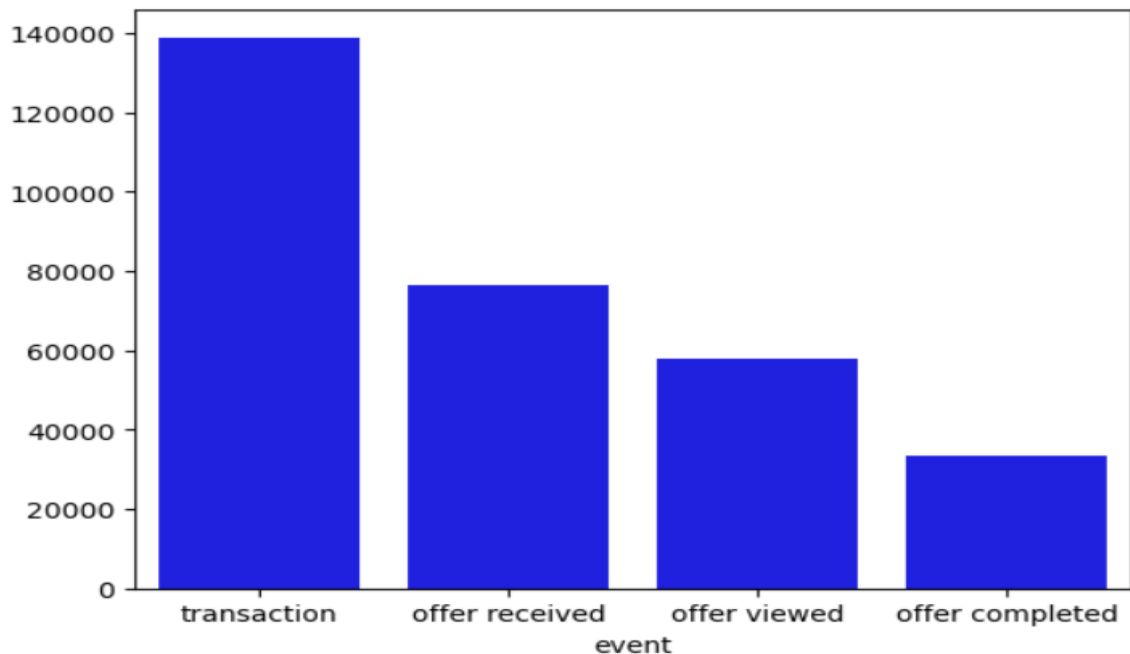


Fig 1. Bar plot of the event column of the transcript dataset.

The bar chart illustrates the frequency distribution of each value. Based on the problem statement, only the last three values are of primary interest. It was observed that many of the “transaction” values represent repetitions of offer completions. Therefore, these “transaction” values will be excluded in subsequent steps. Furthermore, the depiction reveals that there is a considerable amount of data available for model development, as “offer completion” has a count exceeding 40,000. However, this number is expected to decrease following the data preprocessing phase.

During data preprocessing, all customers with incomplete information (i.e., missing values for gender and income) were removed from the profile dataset. This decision was made because customers with incomplete information constituted only about 13% of the total profiles, which was deemed minimal. Moreover, the algorithm selected for training (XGBoost) does not effectively

handle missing values. Subsequently, all transactions involving customers with incomplete demographic information were removed from the “transcript” dataset. This adjustment resulted in the removal of approximately 11% of the transactions, ensuring that the integrity of the dataset remained largely preserved.

To reiterate, the objective of this project is to develop a model that can predict whether customers will complete offers. However, it should be noted that customers can only complete offers if they first view them. Therefore, in the next step, individuals who did not view any offers were excluded from the dataset. This was achieved using the pandas library to group events by customer and then verifying whether the individual’s event history included an “offer viewed” entry.

As previously mentioned in the project overview, informational offers were discarded from the dataset. This decision was based on the observation that informational offers are never marked as completed in the dataset, as confirmed through code executed earlier. Thus, there is no way to determine whether a customer responded to such offers. Out of the nine offers listed in the “portfolio” dataset, two were identified as informational offers. Using their corresponding IDs, all informational offers were removed from the “transaction” dataset. Subsequently, all “transaction” events were eliminated from the “transcript” dataset, leaving only transactions related to offers.

The next significant transformation involved merging the “transcript” and “portfolio” datasets. The customer ID was set as an index, which facilitated this integration. The resulting dataset now includes “event,” “time,” “offer_id,” and “duration” as its primary features.

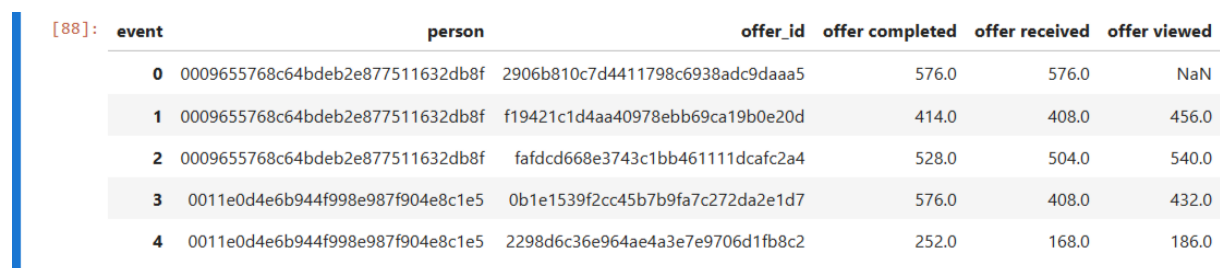
person	event	time	offer_id	duration
0009655768c64bdeb2e877511632db8f	offer received	408	f19421c1d4aa40978ebb69ca19b0e20d	5
0009655768c64bdeb2e877511632db8f	offer completed	414	f19421c1d4aa40978ebb69ca19b0e20d	5
0009655768c64bdeb2e877511632db8f	offer viewed	456	f19421c1d4aa40978ebb69ca19b0e20d	5
0009655768c64bdeb2e877511632db8f	offer received	504	fafdc668e3743c1bb461111dcafc2a4	10
0009655768c64bdeb2e877511632db8f	offer completed	528	fafdc668e3743c1bb461111dcafc2a4	10

Fig 2: Updated dataset

An exploratory analysis of the dataset revealed that an individual could receive the same offer multiple times. This situation could potentially introduce ambiguity, as it might lead to incorrect assumptions that an offer marked as viewed or completed was actually an expired offer. Therefore, a filtering process was implemented to address this issue. Specifically, for each row in the dataset, if an “offer received” event for a particular “offer_id” and “individual” did not have a corresponding “offer viewed” event before the specified expiry time, the row containing that “offer received” event was removed.

To implement this solution, several preliminary steps were required. First, to ensure consistency with the “time” column, the “duration” column was converted from days to hours. Subsequently, a new column named “time of expiry” was created. This column utilized the “time” and “duration” values to calculate when each offer would expire. However, this “time of expiry” column was only populated for rows corresponding to the “offer received” events, while other event types were assigned a “NaN” value. Based on these calculated expiry times, the rows containing expired, unread “offer received” events were removed from the dataset.

The next significant preprocessing step involved restructuring the dataframe using the “pivot_table” method. Through this transformation, only the “person” and “offer_id” columns were retained, while the distinct events in the “event” column were converted into separate features. The values within these features indicated the specific time at which each event occurred. The resulting dataframe, as shown in the figure below, illustrates this new structure.



	event	person	offer_id	offer completed	offer received	offer viewed
0	0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5		576.0	576.0	NaN
1	0009655768c64bdeb2e877511632db8f	f19421c1d4aa40978ebb69ca19b0e20d		414.0	408.0	456.0
2	0009655768c64bdeb2e877511632db8f	fafcd668e3743c1bb461111dcafc2a4		528.0	504.0	540.0
3	0011e0d4e6b944f998e987f904e8c1e5	0b1e1539f2cc45b7b9fa7c272da2e1d7		576.0	408.0	432.0
4	0011e0d4e6b944f998e987f904e8c1e5	2298d6c36e964ae4a3e7e9706d1fb8c2		252.0	168.0	186.0

Figure 4

The image above indicates that there is a missing value in the “offer viewed” column for one of the features. To create a suitable dataset for the intended analysis, it is necessary for every single offer to have been viewed. Therefore, all rows with missing values in this column were removed.

Additionally, any row where the offer was completed before it was viewed was deleted to maintain logical consistency.

Since every remaining row corresponds to an offer that has been viewed, the “offer viewed” column was subsequently dropped, as it no longer provides meaningful information for the model. At this stage, all values in the “offer completed” column were converted to 1, while null values were left as 0. This column will serve as the target variable (label) since the objective is to predict whether a customer will complete an offer. Moreover, the “duration” column from the “portfolio” dataset was merged into the combined dataset using “offer_id” as the key.

The final step in the data cleaning and preprocessing process was to incorporate the remaining columns from the portfolio and profile datasets. To achieve this, it was necessary to ensure that the common features across the datasets had consistent naming conventions. For instance, the “id” column in the profile dataset was renamed to “person” to align with the existing combined dataset. Subsequently, the following columns from the profile dataset were merged into the combined dataset: “gender,” “age,” “became_member_on,” and “income.” From the portfolio dataset, the following columns were added: “offer_type,” “difficulty,” “channels,” and “reward.”

Once these columns were merged, the “person” and “offer_id” columns were removed, as they are irrelevant features for a machine learning model, providing no numerical value beyond identification. The figure below presents the final combined dataset, named “pivot_events,” before proceeding to the feature engineering phase.

[115]:

	offer_completed	duration	gender	age	became_member_on	income	offer_type	difficulty	channels	reward
0	1	240	O	40	20180109	57000.0	discount	20	[web, email]	5
1	1	168	O	40	20180109	57000.0	discount	7	[web, email, mobile, social]	3
2	1	168	O	40	20180109	57000.0	bogo	5	[web, email, mobile]	5
3	1	120	F	59	20160304	90000.0	bogo	10	[web, email, mobile, social]	10
4	1	240	F	59	20160304	90000.0	discount	10	[web, email, mobile, social]	2
5	1	168	F	24	20161111	60000.0	discount	7	[web, email, mobile, social]	3
6	1	168	F	24	20161111	60000.0	bogo	5	[web, email, mobile]	5
7	1	120	F	24	20161111	60000.0	bogo	5	[web, email, mobile, social]	5
8	1	240	F	26	20170621	73000.0	discount	10	[web, email, mobile, social]	2
9	1	240	F	19	20160809	65000.0	discount	10	[web, email, mobile, social]	2

Figure 5: First few rows of the final dataset before feature engineering.

The “offer completed” column serves as the target label for the dataset. Upon executing the “value_counts” method on this column, it was observed that there are approximately 19,000 positive (true) values and 10,000 negative (false) values. Although the dataset is somewhat imbalanced, there are a sufficient number of instances in each class to effectively train a model. However, due to this imbalance, accuracy is not an appropriate evaluation metric, as it would be biased towards the majority class. Consequently, the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) has been chosen as the preferred evaluation metric, as it provides a more balanced assessment of the model’s performance.

[117]:

	offer completed	duration	age	became_member_on	income	difficulty	reward
count	29511.000000	29511.000000	29511.000000	2.951100e+04	29511.000000	29511.000000	29511.000000
mean	0.648978	168.681509	54.441259	2.016698e+07	65353.901935	8.911592	5.480804
std	0.477298	43.188207	17.169409	1.196124e+04	21008.132753	3.431670	3.200257
min	0.000000	120.000000	18.000000	2.013073e+07	30000.000000	5.000000	2.000000
25%	0.000000	120.000000	43.000000	2.016052e+07	50000.000000	7.000000	3.000000
50%	1.000000	168.000000	55.000000	2.017081e+07	64000.000000	10.000000	5.000000
75%	1.000000	168.000000	66.000000	2.018010e+07	79000.000000	10.000000	10.000000
max	1.000000	240.000000	101.000000	2.018073e+07	120000.000000	20.000000	10.000000

Figure 6: Statistical description of the numerical features of the dataset

The figure above provides a statistical summary of the numerical features in the dataset. As observed in the “duration,” “age,” “income,” “difficulty,” and “reward” columns, the mean and the 50th percentile (median) values are in close proximity. This indicates that the data distribution for these features is approximately symmetrical, suggesting that there is no significant skewness. Consequently, logarithmic scaling or other transformations to address skewness will not be required.

FEATURE ENGINEERING

The feature engineering process began by applying one-hot encoding to the “gender” feature. This feature contains three distinct values: “F,” “M,” and “O.” Each of these values was separated into its own individual column. Subsequently, the “became_member_on” column was converted to a datetime format using the pandas “to_datetime” method. This conversion enabled the extraction of the “year,” “month,” and “day” components from the datetime feature using standard datetime methods.

The “day” feature was discarded, as it was determined that the specific day of the month is unlikely to have a significant impact on the model’s predictions. Additionally, creating one-hot encoded columns for 31 possible values would unnecessarily increase the dimensionality of the dataset. The “year” feature was retained in its current form without encoding, while the “month” feature was one-hot encoded to account for the cyclical nature of months, which reset from 12 to 1 at the start of each year.

Next, for the “offer_type” feature, which only includes two values: “discount” and “bogo,” numerical encoding was applied using “0” and “1,” respectively. In the “channels” column, each entry consists of a list of communication channels associated with each offer. To simplify interpretation, each distinct channel was assigned its own column, thereby creating separate binary columns for each possible channel. This approach ensures that the information is presented in a form that is more easily interpretable by the model.

Finally, all numerical features were scaled to values between 0 and 1. Although scaling is not strictly required for XGBoost models, it was performed to ensure uniformity across features, thereby making the data easier for the model to process.

TRAINING AND TUNING

CREATION OF TRAINING FILES

We’ve finally completed the analysis stage. Now, the dataset will have to be prepared for training. First thing that was to split the data into three parts(train, valid, test) with the majority of the data

going into the training set. Next, these files were saved in “csv” formats and saved in S3 bucket for training.

INITIATION OF THE TUNING JOB

Now that the dataset's been uploaded to S3, the tuning job can be configured. Before that, I need to select which model will be used for this problem. As stated earlier, I'll be using Sagemaker's in-built XGBoost algorithm. The reason for this is XGBoost is the most complex built in model on Sagemaker that's not a neural network. The latter reason will help us understand how our features led to the results which is something that can't be done using Neural Networks. Also Sagemaker fully manages the training process so it's relatively easy to train, deploy and monitor. We'll be using a hyperparameter tuning job to train. This will enable different training jobs to be run with different hyperparameters so the best model could be chosen at the end. To perform this tuning job, there are objects that have to be defined. These objects contain the arguments that will be used to run the job. These arguments include the hyperparameters and the range of values for tuning. According to Sagemaker's XGBoost documentation, there are a lot of hyperparameters but only the following really have a noticeable effect on training. These are: “eta”, “alpha”, “min_child_weight”, “subsample”, and “num_round”. The range of values to tune were also documented in the documentation. One hundred training jobs with a maximum of 2 parallel jobs using a Bayesian strategy was selected. This ensures that we get a lot of jobs run to exhaust as much of the dataset as possible. Also, 2 parallel jobs were selected using a Bayesian strategy to make sure that the tuning job isn't just tuning randomly but is trying to find the best set of hyperparameters. The “MetricName” was set to “validation:auc” confirming that we're comparing the training jobs based on their AUC performance in the validation data. Also, “Maximise” was set in order to choose the model that has the best AUC.

Next object that was defined was the “training_job_definition” object. The training image was selected which is XGBoost with a framework version of 1.7-1. The paths to our datasets were also selected here. The instance type chosen here was “ml.c4.2xlarge”. This a compute intensive instance which does not require GPU acceleration which is perfect for our use case since we're not training a neural network. The output folder was also selected for training. The tuning job was run and in the image below, some of the results of the training jobs are shown:

Training jobs						
Sorting by objective metric value will display only jobs that have metric values.						
<div> <input type="button" value="Refresh"/> <input type="button" value="View logs"/> <input type="button" value="View instance metrics"/> <input type="button" value="Stop"/> <input type="button" value="Create model"/> </div> <div> <input type="text" value="Search training jobs"/> <div> <input type="button" value="Previous"/> 1 <input type="button" value="Next"/> <input type="button" value="Settings"/> </div> </div>						
	Name	Status	Final objective metric value	Creation time	Training Duration	
<input type="radio"/>	XGBoostTuningJobv5-100-191dd89e	Completed	0.7911199927330017	9/27/2024, 9:08:02 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-099-6e6c82e6	Completed	0.7892400026321411	9/27/2024, 9:07:11 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-098-d388e880	Completed	0.7896400094032288	9/27/2024, 9:06:42 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-097-d9194047	Completed	0.7898600101470947	9/27/2024, 9:06:05 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-096-05335f27	Completed	0.7898300290107727	9/27/2024, 9:05:38 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-095-f72e3189	Completed	0.7891799807548523	9/27/2024, 9:05:01 PM	1 minute(s)	
<input type="radio"/>	XGBoostTuningJobv5-094-d3f61b4d	Completed	0.7887700200080872	9/27/2024, 9:04:28 PM	1 minute(s)	

Figure 7: Best training jobs from the tuning job

The best training job has an AUC score of 0.79. In essence, AUC is the likelihood that for every pair of positive and negative values that the right label is assigned for each. 0.79 seems a decent enough score for our case and we'll perform batch transform and analysis on the model.

ANALYSIS OF RESULTS

A batch transform job was then run on the test set. An AUC score of 0.8 was gotten which is similar to what we got when the validation set was tested. Next we extract the model and perform SHAP analysis. Below is a plot of the SHAP analysis:

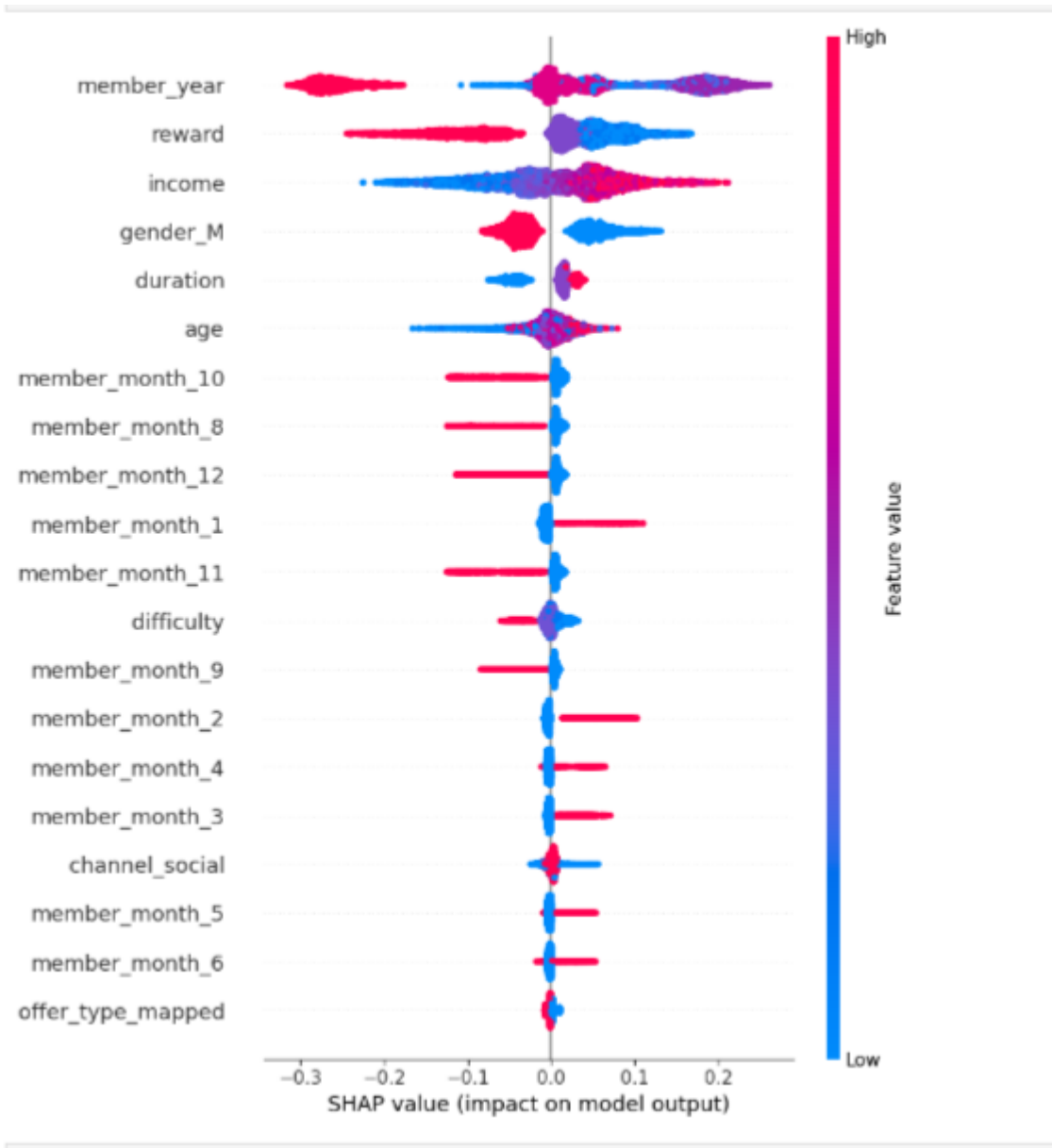


Figure 8: SHAP Analysis plot

Analysis of Features

The SHAP summary plot visualizes how different features impact the model's prediction of whether a customer will complete an offer or not. Features are ranked based on their importance, with the most influential ones at the top. The x-axis shows the SHAP values, representing whether a feature contributes to increasing or decreasing the likelihood of completing an offer.

1. Membership Year (member_year):

This is the most impactful feature. Interestingly, customers who became members earlier (lower member_year values shown in blue) are more likely to complete offers, while newer members (higher member_year values shown in red) are less likely to do so. This suggests that long-term customers tend to engage more with promotions compared to recent members.

2. Reward (reward):

The reward amount associated with the offer has a varied influence. For some customers, higher rewards increase their chances of completing an offer, but for others, it makes little difference. This mixed impact indicates that simply offering a higher reward may not always drive engagement; other factors might be at play, such as the customer's income or spending habits.

3. Income (income):

Lower-income customers (blue) are more likely to complete an offer, as indicated by positive SHAP values, whereas high-income customers (red) are less likely to respond positively to promotions. This suggests that promotions might be more effective for customers with limited disposable income, who might view discounts or rewards as more valuable.

4. Other Features:

Features like gender (gender_M), duration of the offer (duration), and age (age) also play a role but are less influential compared to the top three. Gender appears to influence predictions inconsistently, while offer duration is only moderately impactful. Membership month features (e.g., member_month_10) are relatively less important but might indicate seasonality trends or specific membership patterns.

In conclusion, the top three features—member_year, reward, and income—drive most of the model's decisions. Understanding these relationships can help Starbucks fine-tune their promotions to better engage different customer segments, especially focusing on long-standing members and customers with lower incomes.

CONCLUSION

The project aimed to build a model that can predict whether a Starbucks customer will complete an offer based on demographic and offer-related information. By integrating customer profiles, offer details, and transaction histories, the model provides valuable insights into what drives offer completion.

The analysis highlighted the importance of membership duration, reward amount, and customer income as key factors influencing engagement with offers. Long-term members and customers with lower incomes are more likely to respond positively to promotions. This suggests that tailored marketing strategies focusing on these customer segments could yield higher engagement rates.

The SHAP analysis further revealed nuanced patterns, indicating that simply increasing rewards or extending offer durations may not always be effective. Instead, understanding customer behaviors and preferences at a deeper level is crucial for optimizing promotional strategies.

Overall, the project successfully demonstrated how machine learning can help personalize marketing campaigns, offering Starbucks a data-driven approach to enhancing customer loyalty and satisfaction.

REFERENCES

1. Jacinta Dias (2024). 6 Successful Marketing Strategies of Starbucks. Retrieved from <https://iimskills.com/marketing-strategy-of-starbucks/>
2. Yixuan Zhu (2024). Leveraging Social Media Marketing: A Case Study of Starbucks Digital Success. Retrieved from https://www.researchgate.net/publication/382586535_Leveraging_Social_Media_Marketing_A_Case_Study_of_Starbucks_Digital_Success