Proposal

Evaluating Customer Engagement Programs at Starbucks

Domain Background

Many companies use incentive programs to increase sales, raise brand awareness, and encourage customer loyalty.
The design of the program often depends on the type of customer it is targeting – new (coupons), infrequent (discounts), regular (rewards proportional to purchase value, eg. Amtrak Rewards) , frequent (membership-based programs, eg. Amazon Prime). The type of offer presented also depends on the product being sold (single product, limited variety, large selection)

Problem Statement

Instead of sending all app users the same offer, we would like to target our customers carefully – by sending them offers that they are likely to respond to, else they could stop using the app. Also, some users are regulars – and they will spend anyway. They should not receive discounts.

We need to predict the offers a person should receive based on age, gender, income, and purchasing history. This prediction is quantifiable, and measurable, because it will include an expected spend for the duration of the offer. This can be compared to the true spend.

Datasets and Inputs

The data comes from usage of the Starbucks app. Three datasets are available:

1. Portfolio:

The Portfolio table has 10 rows corresponding to 10 types of offers. The fields are:

- id (string) - offer Id
- offer_type (string) - BOGO, discount, informational (just an ad, no value)
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) - time for offer to be open, in days

- channels (list of strings) – web, email, mobile, social

For example – a discount type offer, where, if $20 (difficulty) is spent within 10 days (duration), a reward of $5 will be given. It will be sent out by web or email channels, but not be mobile or social networks.

2. Profile:

In this table, each row corresponds to a customer and his or her demographic information – age, gender and income. There are 17,000 customers. Complete records are available for 14,825 individuals. In this project, these variables – age, gender and income - will be used to predict the offer a customer should receive.

3. Transcript:

This table contains events for all customers over a period of 30 days (720 hours). An event can be

i. a transaction, where a customer makes a purchase in dollars
ii. an offer event – offer received, offer viewed, or offer completed
   The other field is “time” of the event, described above, in hours since the time of the 30-day program

Solution Statement

The model must predict the offer a customer should receive from a set of 10 offers. This is a multi-class prediction. The inputs – age, gender, and income, are structured data. So a boosted Decision Tree or Random Forest will work well. These models are explainable.

Benchmark Model

For multiclass prediction, the multinomial logistic model can be used as a standard benchmark model. It is frequently more accurate than the naive majority-class predictor. Its components are additive, so it is easy to explain. It is widely used in a lot of fields from public health to marketing.

Evaluation Metrics

We will compare predicted offers against actual offers for the test cases. We will estimate the accuracy of completion of the offer, and compare spend during the offer duration. Accuracy is a good evaluation metric because 1) The data is balanced. There are 1500-4500 completions for each offer type. 2) The cost of a false positive or false negative is low, unlike in healthcare and fraud detection problems. So precision & recall metrics are not necessarily better.

Project Design

In this project, we're predicting offers that an individual will like. Because the customer will receive many (3-6) offers during the 30-day study period, the number of records will be a small multiple of the number of customers (17,000).

1 Preprocessing:

Useful data must be extracted from the Portfolio (offer types), Profile (customer info) and Transcript (offers & spend) tables. For each customer in the Profile table, we will extract offers received, viewed, completed & amount spent from the Transcript table. This will become a customer spend (per offer received) DataFrame (for each customer) – called cust_spend_df

A single customer can receive multiple offers – even many of the same offer type. So we will copy the row for the customer from the Profile table, and duplicate it – once for each offer received row in the cust_spend_df. This is the cust_profile_df

For each offer received, we will get offer attributes (type, difficulty, reward, and duration) from the Portfolio table. We'll also add an "offer_label" column to quickly identify an offer, instead of using the offer_id in graphs and tables. For example, "d100207" is a discount type offer, with a difficulty of 7, a reward of 2, and a duration of 7 days. This cleaned table will become the cust_offers_df

These three customer-specific dataframes will be stacked horizontally, to become the cust_offers_received_df. Then the dataframes for ALL customers will be stacked vertically. Each row of the offers_rec_df is an offer received, and its calculated attributes (including total spend). Its dimensions are (68574, 18). The 18 variables are ['time_received', 'offer_rec_id', 'time_since_first_offer', 'has_viewed', 'time_viewed', 'has_completed', 'time_completed', 'spend', 'gender', 'age', 'cust_id', 'became_member_on', 'income', 'offer_label', 'offer_type', 'difficulty', 'reward', 'duration']

2 Data Exploration:

A visual exploration will be conducted to understand the factors that lead to offer completion (eg spend per offer, income, time between receiving the offer and viewing it). We'll also explore offer preference – why a customer prefers one offer over another – to discover insights into customer behavior.

3 Data Modeling:

A model will be trained using XgBoost in Amazon SageMaker or AdaBoost in Scikit Learn (decision tree models). The data is structured. These models tend to perform better on such data, while keeping interpretation simple

4 Model Benchmarking:

The trained model will be compared against the Multinomial Logistic Model, which is the industry standard in many fields.