# Starbucks Promotional Response Prediction Using Machine Learning

## 1. Domain Background

Digital loyalty programs have become a central strategy in modern marketing, allowing companies to deliver targeted promotions and measure customer engagement at scale. Starbucks' mobile rewards program is one of the most influential examples—customers receive promotional offers such as buy-one-get-one (BOGO), discounts, and informational ads, each delivered through different channels (email, web, mobile, social).

The dataset for this project is **a simulated dataset designed to reflect realistic behavioral patterns rather than random synthetic noise**. Different types of customers exhibit different tendencies:

- Some respond strongly to certain offers
- Some ignore or **negatively** respond to marketing campaigns
- Some make purchases regardless of promotional influence
- Some complete offers without ever viewing them (i.e., not influenced)

---

## 2. Problem Statement

Customers vary widely in how they respond to offers. Sending the wrong offer can waste marketing budget, reduce margins, or annoy customers—including those who prefer **not** to receive any promotions.

**Goal:**
Predict whether a customer will **respond positively** to a given promotional offer.

A "response" is defined as:
**Offer viewed**, and
**Offer completed**,
**Within the offer's validity period**

This definition reflects true marketing influence rather than simple event co-occurrence.

The problem is:

- measurable (timestamped events),
- actionable (send / do not send offer),
- and realistic in its causal structure.

---

# 3. Datasets and Inputs

The project uses the three provided Starbucks JSON files:

### 1. portfolio.json — Offer Metadata

- `id`: offer ID
- `offer_type`: bogo / discount / informational
- `difficulty`: minimum spend required
- `reward`: incentive provided
- `duration`: validity window
- `channels`: list of delivery channels

### 2. profile.json — Customer Demographics

- `id`: customer ID
- `age`, `gender`, `income`
- `became_member_on`: membership start date

### 3. transcript.json — Event Log

Records include:

- `offer received`
- `offer viewed`
- `offer completed`
- `transaction`

Each entry contains:

- `person` = customer ID
- `time` = hours since test start
- `value` = offer_id, amount spent, or reward

This dataset enables:

- complete reconstruction of offer timelines
- causal labeling based on viewing + completing
- detection of customers who complete without being influenced
- identification of customers who prefer no offers

---

## 4. Solution Statement

The solution is to build a supervised machine learning system that predicts whether a customer will positively respond to a promotional offer based on their demographics, past behavior, and the characteristics of the offer itself. By combining event-level interaction logs with customer and offer metadata, the model learns which segments are likely to engage with specific promotions and which segments should not be targeted. This enables Starbucks to optimize promotional spend, personalize marketing strategies, reduce customer fatigue from irrelevant offers, and improve overall campaign effectiveness through data-driven decision making.

---

# 5. Benchmark Model

The benchmark is **Logistic Regression**, chosen because:

- It is widely used in traditional marketing response modeling
- It provides interpretable coefficients
- It establishes a performance baseline

Observed benchmark performance:
**Accuracy: ~0.75**
**ROC-AUC: ~0.839**

This serves as the threshold that more sophisticated models must beat.

---

# 6. Evaluation Metrics

### Primary Metric — ROC-AUC

- Measures ranking quality
- Works well for imbalanced classes
- Ideal for marketing response prediction

### Secondary Metrics — Precision, Recall, F1

These help evaluate:

- How many true responders the model captures
- How many false responders the model incorrectly predicts
- Trade-offs between marketing reach and wasted promotions

---

# 7. Project Design

## Step 1 — EDA & Data Cleaning

- Examine demographic distributions
- Expand transcript `"value"` field
- Investigate event frequencies
- Handle missing ages and genders
- Identify unrealistic ages (e.g., age 118 → missing)

## Step 2 — Offer Labeling Logic

For each offer instance:

- Determine offer window based on `duration * 24 hours`
- Check if offer was **viewed** within window
- Check if offer was **completed** within window
- Determine:
  - positive responders
  - negative responders (viewed but not completed)
  - uninfluenced completers (completed without viewing)
  - offer-irrelevant purchasers

## Step 3 — Feature Engineering

- One-hot encode channels

- Encode offer types
- Membership tenure
- Transaction behavior
- Difficulty, reward, and duration

### Step 4 — Modeling

- Train/Test/Validation split
- Logistic Regression
- Random Forest
- AutoGluon with bagging & stacking
- Select the best model (AutoGluon WeightedEnsemble_L3)

### Step 5 — Deployment Artifacts

- Save the trained AutoGluon predictor
- Optionally upload to S3
- Demonstrate inference using:
  - local notebook prediction
  - model reload from disk

### Step 6 — Insight Extraction

Interpret model outputs to identify:

- which demographic groups respond most
- which offers work best for which customers
- segments for which *no offer is optimal*