Personalized Offer Response Prediction Starbucks Project Report

Name: Y Zhou

1. Project Overview

Personalization is essential in modern marketing. Starbucks is delivering targeted promotional offers to their users. These offers range from discounts to buy-one-get-one (BOGO) deals and informational ads. Understanding which users are more likely to respond to which type of offer enables businesses to improve engagement, increase revenue, and minimize promotional costs.

This project uses a Starbucks dataset to build a machine learning model that predicts whether a user will complete a given offer. I integrated customer demographics, historical behavior, and offer characteristics to create a personalized marketing recommendation system.

2. Problem Statement

My goal is to predict the likelihood that a user will respond to and complete a given promotional offer. This involves analyzing:

- Demographic features (age, gender, income, membership length)
- Offer features (type, difficulty, reward, duration)
- Behavioral data (transactions amount, offer viewed, offer received)

I frame this as a binary classification problem: for each offer received, will the user complete it or not?

3. Evaluation Metrics

Given potential class imbalance and business cost implications, the main metrics are:

- **Precision/Recall**: Important for minimizing marketing waste (false positives) and capturing real responders (true positives).
- **F1-score**: Balances precision and recall.

- **ROC-AUC**: Measures classifier's ability to distinguish between classes.

4. Data Exploration

The dataset consists of three JSON files:

- **portfolio.json**: Details of 10 offers (reward, duration, difficulty, type).
- **profile.json**: Information on 17,000 users (age, gender, income, join date).
- **transcript.json**: 306,648 user events (transactions, offer received/viewed/completed).

I identified and handled the following:

- Invalid ages (118) were removed.
- Missing incomes imputed with median.
- Missing genders filled as 'Unknown'.
- Timestamps converted to datetime.

Key Insights (illustrated by plots below)

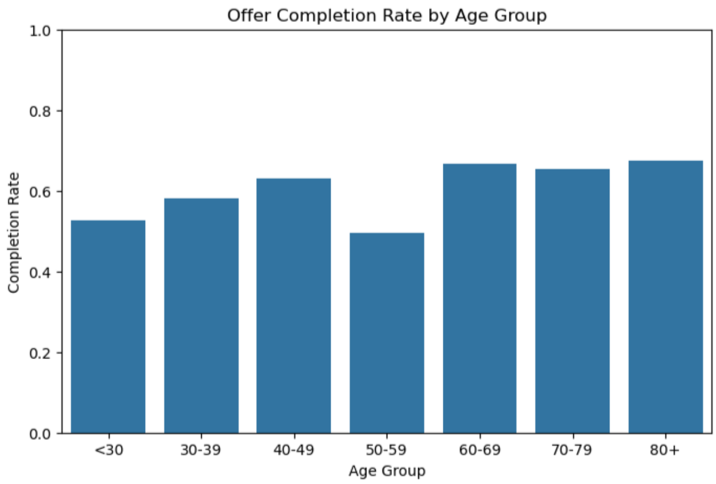
- Males have a relatively lower offer completion rate compared with females and other genders.
- Elder customers tend to have a higher offer completion rate.
- Customers with higher income have relatively higher completion rate.

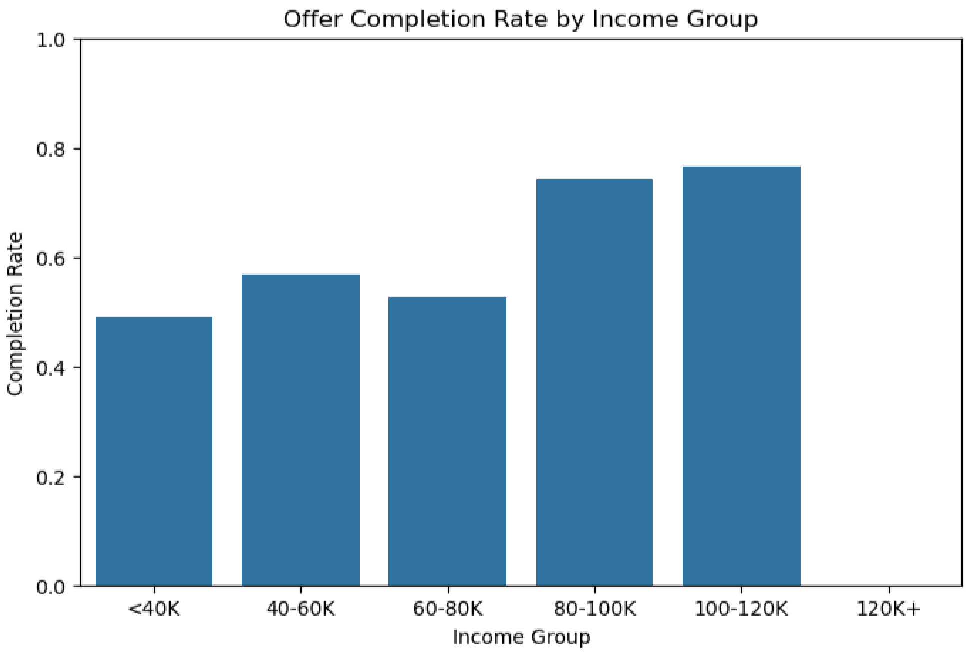
5. Visualizations

Several visualizations helped my analysis.

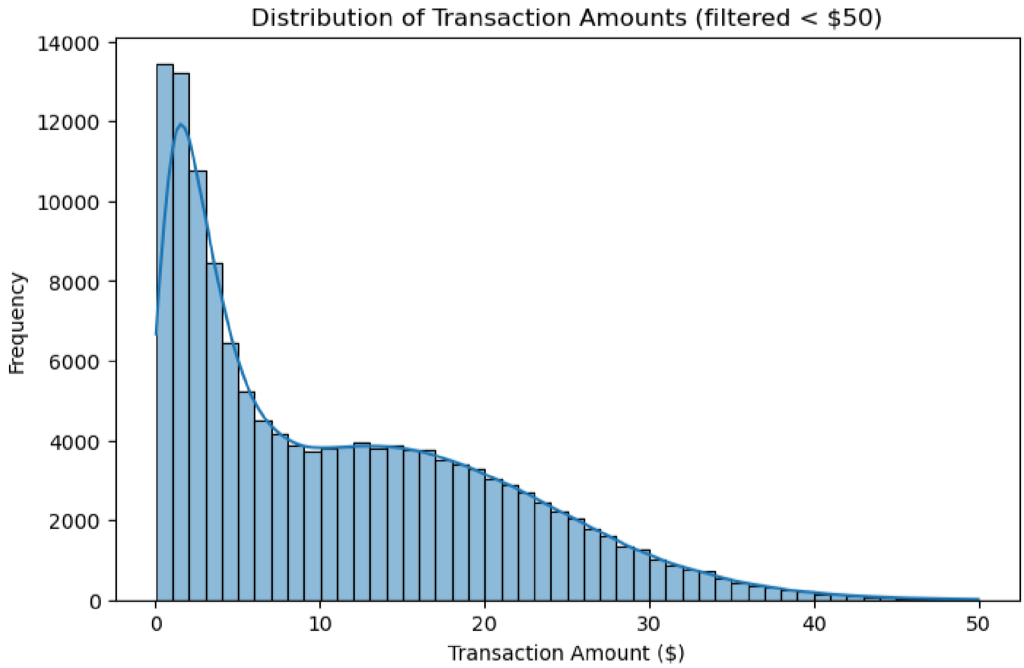
- Completion rates by age and income groups


Offer Completion Rate by Gender
Completion Rate
1.0
0.8
0.6
0.4
0.2
0.0
F
M
O
Unknown
Gender


Offer Completion Rate by Age Group
Completion Rate
1.0
0.8
0.6
0.4
0.2
0.0
<30
30-39
40-49
50-59
60-69
70-79
80+
Age Group

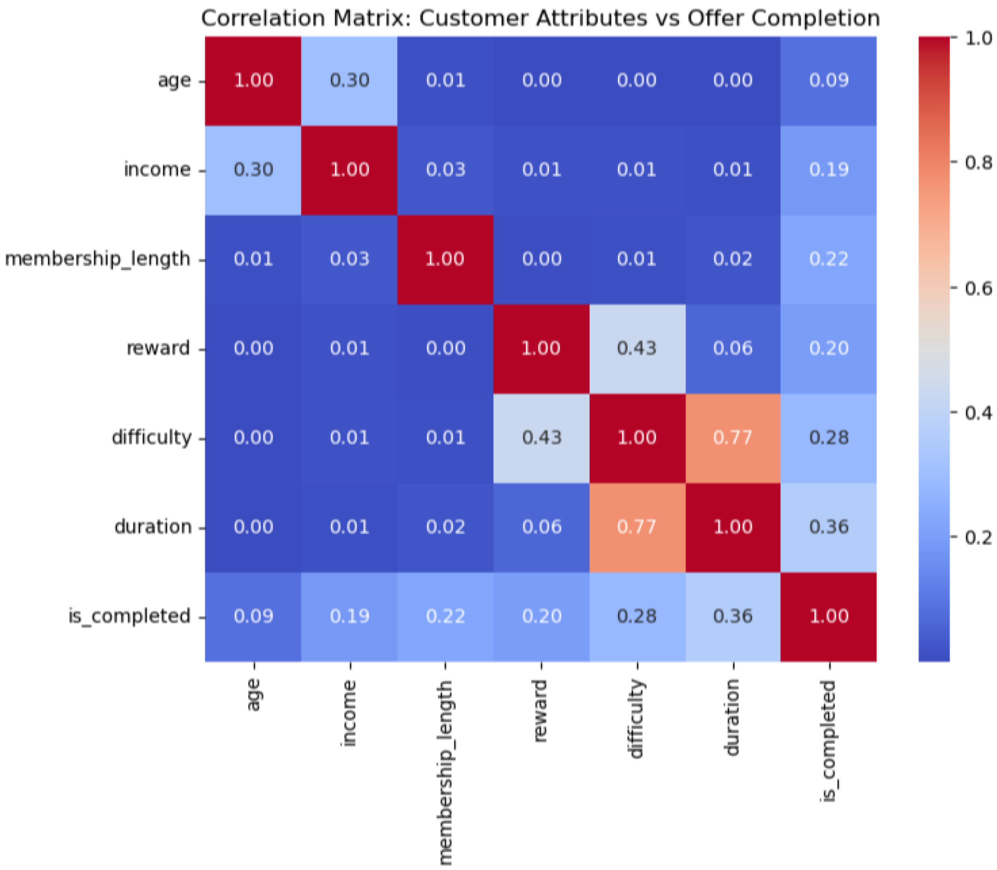


- Transaction frequency by transaction amount



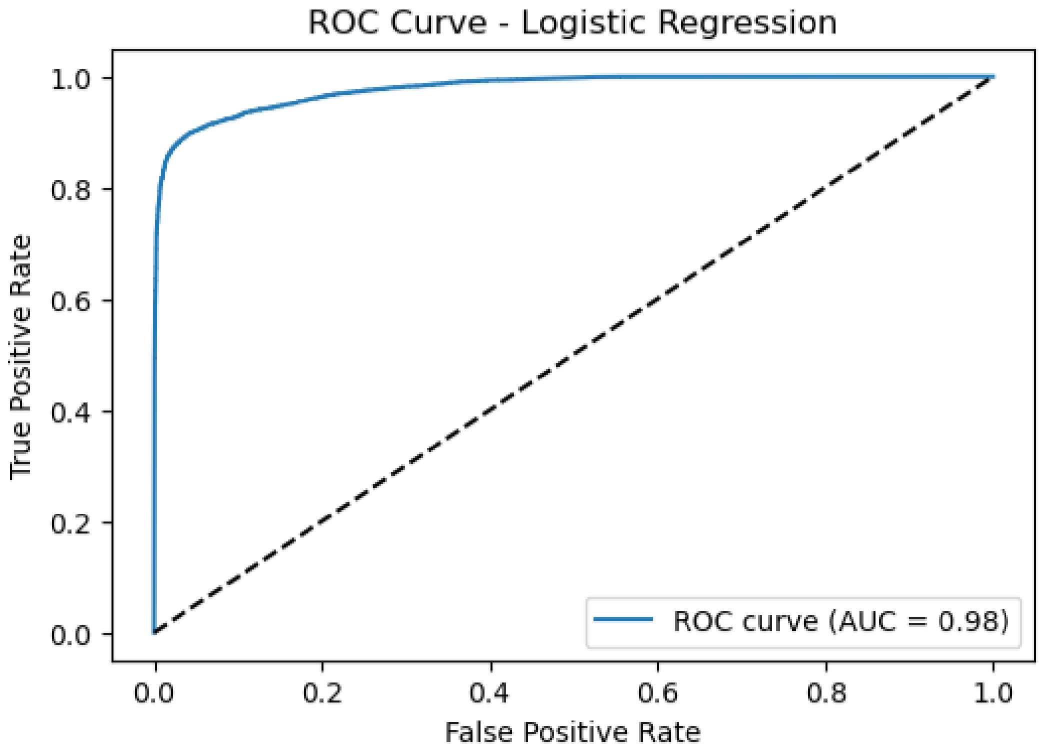
- Correlation matrix among customer attributes and offer completion

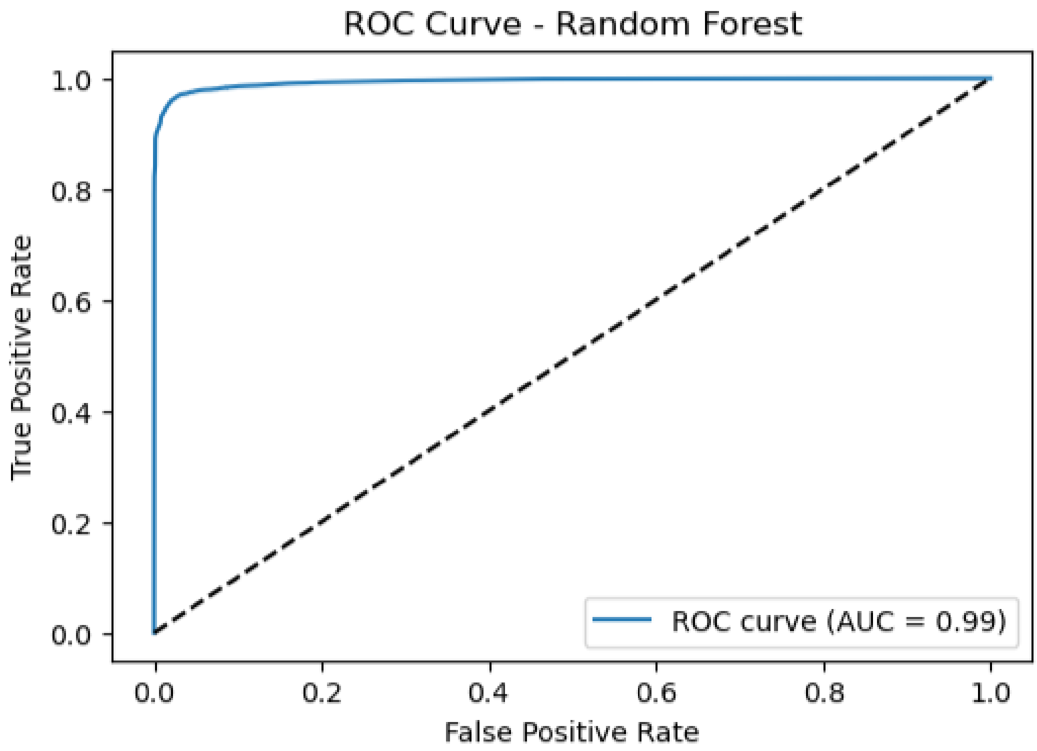
  Plot below shows that membership_length, difficulty, and duration are possible attributes that help to infer offer completion.

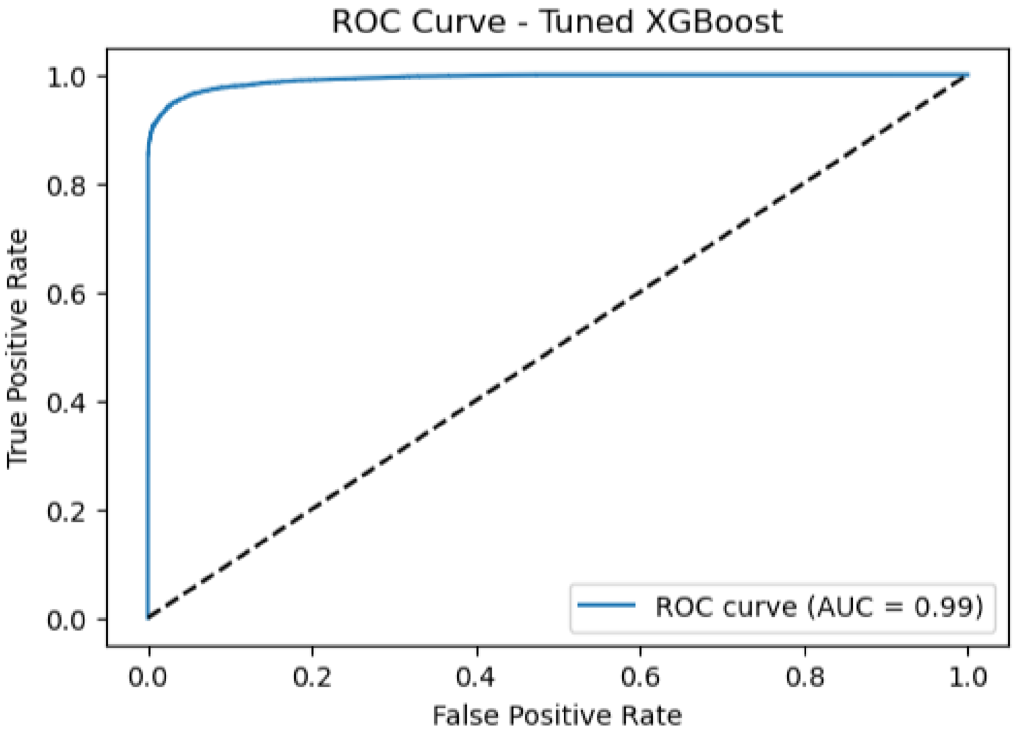


- ROC curves for Benchmark, Random Forest, and XGBoost models

  All models perform well in predicting customer response to offers, among which Random Forest model has a highest ROC-AUC score. Detailed analysis will be illustrated in later sections.


ROC Curve - Logistic Regression
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
ROC curve (AUC = 0.98)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate


ROC Curve - Random Forest
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
ROC curve (AUC = 0.99)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate



6. Feature Engineering

- Merged portfolio, profile, and transcript.
- Created offer lifecycle (received, viewed, completed)
- Calculated time-based features like membership_length
- Engineered behavioral features like transaction_amount during offer window
- One-hot encoded categorical variables (e.g., gender, offer_type)

7. Model Development, Evaluation and Justification

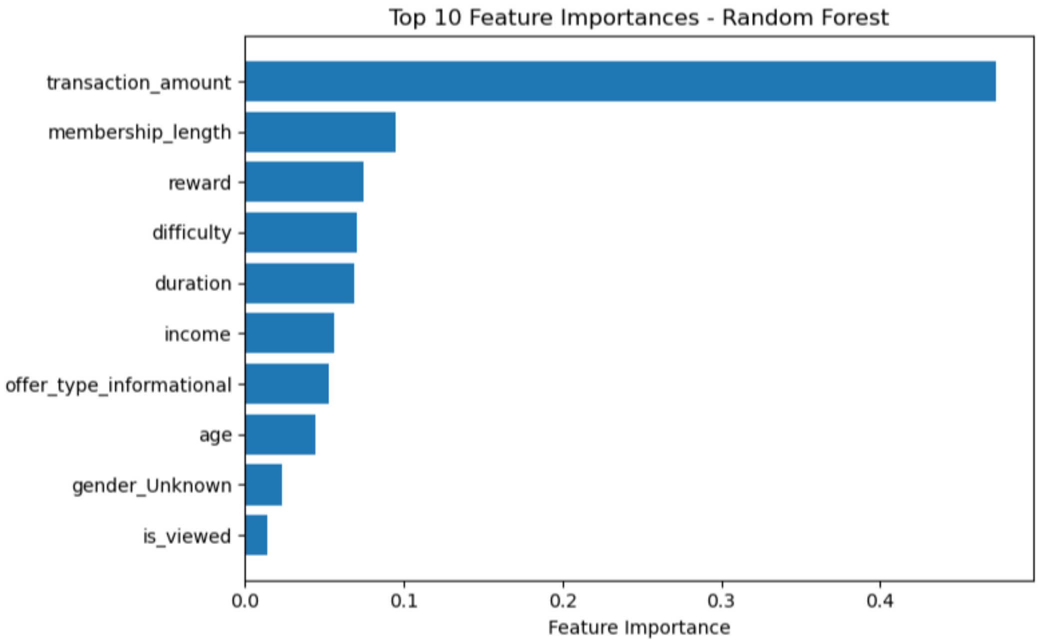
I started with benchmark logistic regression, then improved my performance via Random Forest. I then tuned XGBoost with RandomizedSearchCV, which did not outperform RF. F1-score and ROC curve are my main evaluation metrics for model selection:

- **Logistic Regression**: Benchmark/Baseline (F1 = 0.9286, ROC-AUC = 0.9786)
- **Random Forest**: Final selected model (F1 = 0.9736, ROC-AUC = 0.9941)

- **XGBoost**: Tuned version (F1 = 0.9627, ROC-AUC = 0.9927)

The **F1-score** balances precision and recall, making it particularly useful when class imbalance is a concern. It tells how well the model avoids both false positives and false negatives. On the other hand, **ROC-AUC** measures the model's ability to distinguish between classes across all thresholds. Therefore, both metrics were considered together when selecting the final model. Random Forest was ultimately chosen due to its superior F1-score and ROC-AUC score.

Top Feature Importances in Random Forest model:



8. Business Recommendations

- **Recommendation 1: Personalize Offer Difficulty**

Assign higher-difficulty offers (e.g., BOGO) to users with strong spending histories and longer memberships.

- **Recommendation 2: Target Based on Transaction History**

Provide larger, high-reward offers to high-value segments while using introductory offers for low-value users to encourage engagement.

- **Recommendation 3: Adjust Rewards by Segment**

Larger rewards correlate with higher completion likelihood.

Increase reward amounts for loyal or high-value customers.

9. Summary

This project developed a machine learning solution to predict offer responsiveness for Starbucks. The final Random Forest model achieved strong classification metrics and offers actionable business insights. By leveraging model predictions, Starbucks can:

- Reduce marketing waste by avoiding offer distribution to non-responsive users.
- Increase completion rates and user engagement by offer targeting.
- Enhance customer satisfaction and drive revenue growth by tailoring offer types.

Overall, this model helps to build a personalized, data-driven marketing system.

References

- Udacity Starbucks Capstone Dataset
- scikit-learn documentation
- XGBoost documentation
- GenAI assistance for error correction during model development