Project Report: Starbucks Capstone Challenge

Jared Gan

Udacity – AWS Machine Learning Nanodegree 2023

Introduction:

Starbucks is one of the most popular coffeehouse chains in the world with a presence in over 80 countries. The company has a mobile app that enables customers to pay for their orders, earn rewards, and avail of special offers. The challenge for Starbucks is to analyze customer behavior on the app and identify which demographic groups respond best to which offer types. This project proposes a machine learning solution to predict customer behavior upon receiving offers from Starbucks.

Problem Statement:

The goal of this project is to build a machine learning model that can predict which customers are more likely to make a purchase upon receiving offers. Starbucks sends out different types of offers to its customers, including Buy One Get One (BOGO), discounts, and informational offers. Not all customers receive the same offer, and some customers may not receive any offer during certain weeks. Therefore, the challenge is to analyze the customer behavior on the Starbucks rewards mobile app and identify which demographic groups respond best to which offer types.

Proposed Solution:

To address this challenge, the project proposes the following steps:

1. Exploratory data analysis (EDA): Perform EDA on the provided dataset to understand the relationship between customer demographics, offer types, and customer behavior on the mobile app.
2. Data preprocessing: Preprocess and clean the data, ensuring data integrity and addressing any missing values.
3. Machine learning techniques: Use machine learning techniques to build a predictive model that can determine which customers respond best to offers.
4. Hyperparameter optimization: Optimize the model's hyperparameters using techniques such as grid search or Bayesian optimization.
5. Evaluation: Evaluate the model's performance on the test set and compare it with the performance of other models.

Deliverables:

The deliverables of this project will be a well-documented codebase that includes data preprocessing, EDA, and a predictive model. The model should be able to predict which customers are more likely to make a purchase upon receiving offers. The project will be presented in the form of a technical report that documents the entire project's scope, methodology, results, and conclusions. The report will also include visualizations to help explain the findings and a detailed discussion of the limitations and future scope of the project.

Evaluation Metric:

The F1 score will be used as the primary evaluation metric for this project. The F1 score is a measure of a model's accuracy that considers both precision and recall. By using the F1 score as our primary metric, we can ensure that our model performs well in terms of both precision and recall, which are both important for our project's goals.

Project Design:

1. The project will be divided into the following steps:
2. Set up the environment: Create an Amazon SageMaker notebook instance and set up the required environment.
3. Import necessary libraries: Import the necessary libraries such as pandas, numpy, seaborn, matplotlib, and sagemaker.
4. Data loading and exploration: Load the Starbucks dataset into the notebook instance and explore it to gain a better understanding of the data.
5. Data preprocessing: Preprocess the data by performing tasks such as data cleaning, feature engineering, and feature scaling.
6. Data splitting: Split the data into training, validation, and test sets.
7. Machine learning model training: Train a machine learning model using one of the available algorithms such as XGBoost, Random Forest, or Deep Learning.
8. Hyperparameter optimization: Optimize the model's hyperparameters using techniques such as grid search or Bayesian optimization.
9. Model evaluation: Evaluate the model's performance on the test set and compare it with the performance of other models.

Data Sets

The data is contained in three files:

* portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)

* profile.json - demographic data for each customer

* transcript.json - records for transactions, offers received, offers viewed, and offers completed

Here is the schema and explanation of each variable in the files:

portfolio.json

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- 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)

profile.json

- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

transcript.json

- event (str) - record description (ie transaction, offer received, offer viewed, etc.)
- person (str) - customer id
- time (int) - time in hours since start of test. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record

Perform exploratory data analysis and pre-process the dataset by performing feature engineering, handling missing values, and encoding categorical variables

portfolio.json

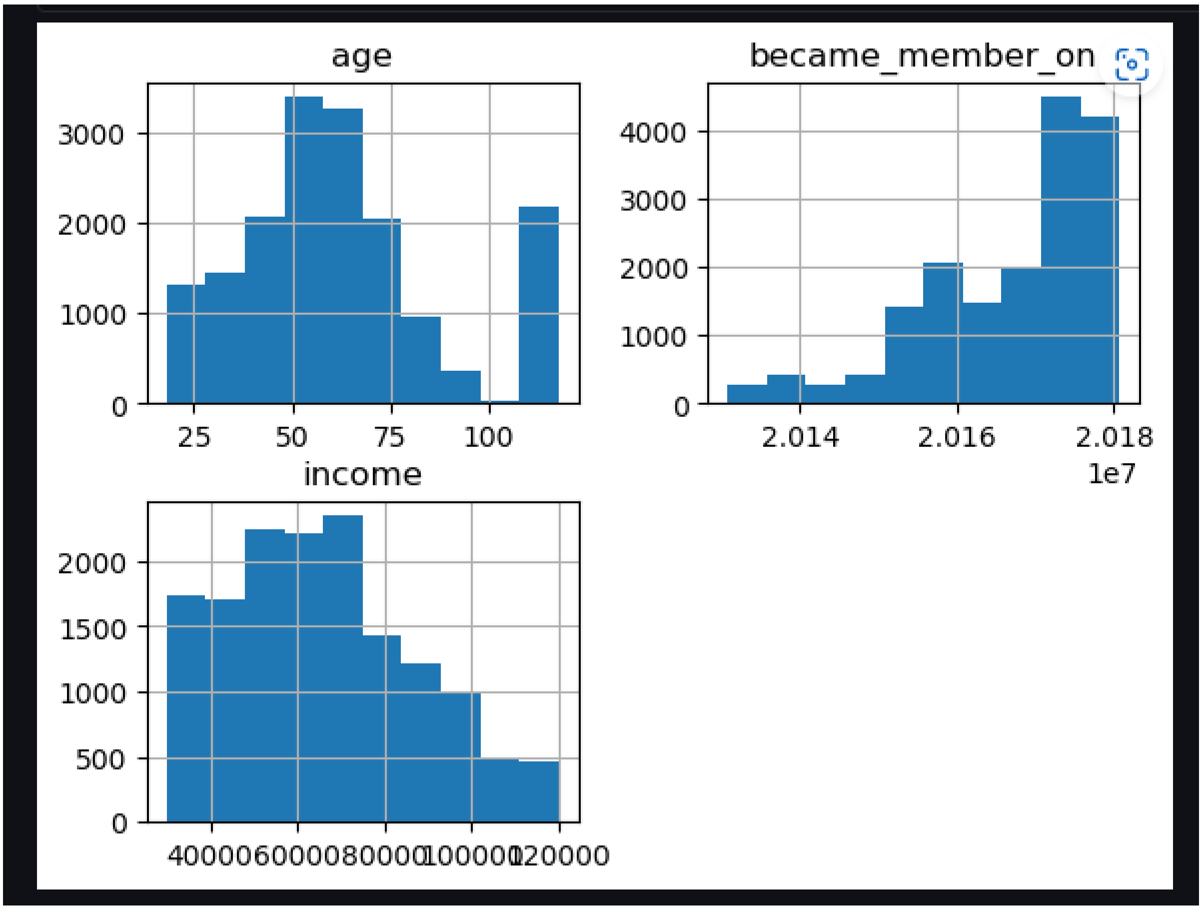
clean version:

No data manipulation required, updated column to “offer id” for dataframe merging.

	reward	channels	difficulty	duration	offer_type	offer id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7

profile.json

Observed irregular data due to age feature as seen in the histogram.
This was most likely caused by customer not providing their age/date of birth and 118 is the default value the app registers
In the process of dropping invalid age data, null gender data was also removed
This implies that customers that did not provide age/date of birth, also did not provide their gender
This then mplies such customers prefer not to disclose their personal data (age/date of birth and gender)

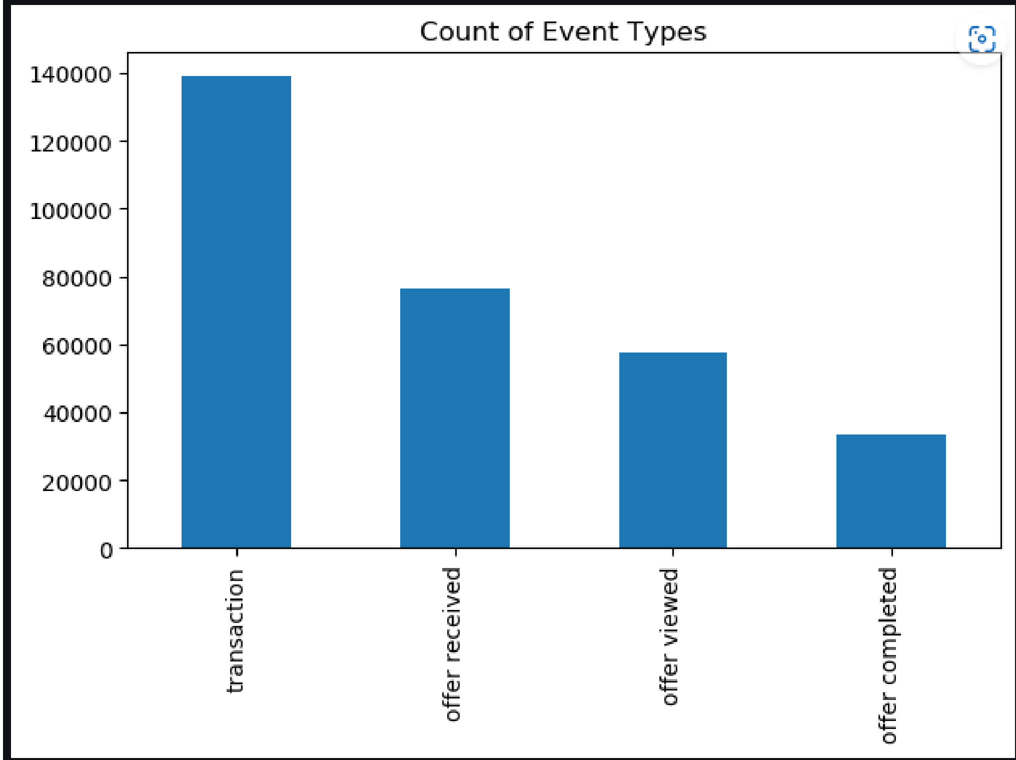


clean version:

	gender	age	customer id	became_member_on	income
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
8	M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
12	M	58	2eeac8d8feae4a8cad5a6af0499a211d	20171111	51000.0

transcript.json

```python
# Check the event data visualy
transcript.event.value_counts().plot.bar(figsize=(7, 4),title="Count of Event Types");
```




```python
# View the data for the first person in the dataset
usr = transcript.loc[0]['person']
transcript[transcript.person == usr]
```

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
15561	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	6
47582	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 19.89}	132
47583	78afa995795e4d85b5d9ceeca43f5fef	offer completed	{'offer_id': '9b98b8c7a33c4b65b9aebfe6a799e6d9...	132
49502	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 17.78}	144
53176	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '5a8bc65990b245e5a138643cd4eb9837'}	168
85291	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	{'offer id': '5a8bc65990b245e5a138643cd4eb9837'}	216
87134	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 19.67}	222
92104	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 29.72}	240
141566	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 23.93}	378
150598	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': 'ae264e3637204a6fb9bb56bc8210ddfd'}	408
163375	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	{'offer id': 'ae264e3637204a6fb9bb56bc8210ddfd'}	408
201572	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}	504
218393	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 21.72}	510
218394	78afa995795e4d85b5d9ceeca43f5fef	offer completed	{'offer_id': 'ae264e3637204a6fb9bb56bc8210ddfd...	510
218395	78afa995795e4d85b5d9ceeca43f5fef	offer completed	{'offer_id': 'f19421c1d4aa40978ebb69ca19b0e20d...	510
230412	78afa995795e4d85b5d9ceeca43f5fef	transaction	{'amount': 26.56}	534
262138	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}	582

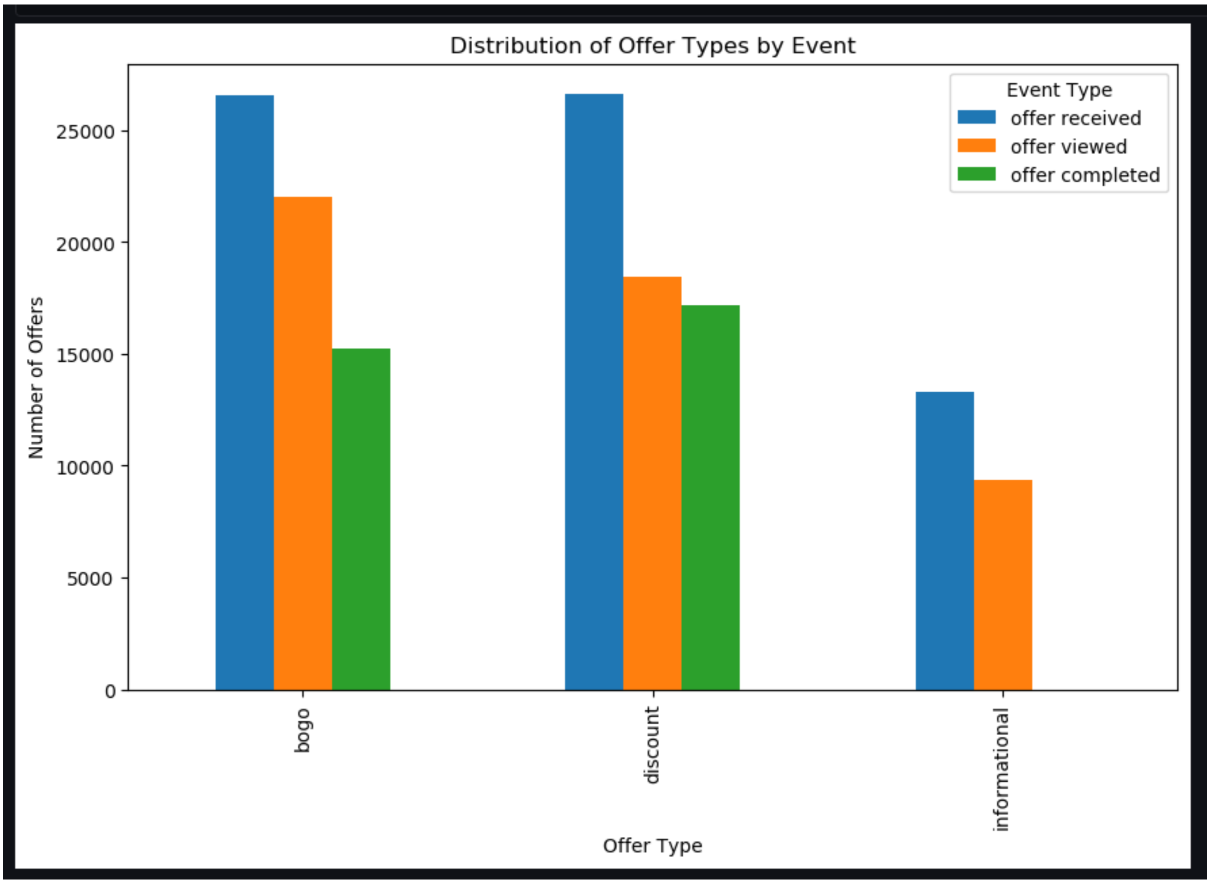
Data indicates that this dataset is a list of historical transactions (multiple transactions per individual)

Dropping and renaming some columns:

	customer id	event	time	offer id
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	9b98b8c7a33c4b65b9aebfe6a799e6d9
1	a03223e636434f42ac4c3df47e8bac43	offer received	0	0b1e1539f2cc45b7b9fa7c272da2e1d7
2	e2127556f4f64592b11af22de27a7932	offer received	0	2906b810c7d4411798c6938adc9daaa5
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	0	fafdcd668e3743c1bb461111dcafc2a4
4	68617ca6246f4fbc85e91a2a49552598	offer received	0	4d5c57ea9a6940dd891ad53e9dbe8da0

Merge the 3 dataframes

	customer id	event	time	offer id	reward	channels	difficulty	duration	offer_type	gender	age	became_member_on	income
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	5	[web, email, mobile]	5	7	bogo	F	75	20170509	100000.0
1	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	6	9b98b8c7a33c4b65b9aebfe6a799e6d9	5	[web, email, mobile]	5	7	bogo	F	75	20170509	100000.0
2	78afa995795e4d85b5d9ceeca43f5fef	offer completed	132	9b98b8c7a33c4b65b9aebfe6a799e6d9	5	[web, email, mobile]	5	7	bogo	F	75	20170509	100000.0
3	78afa995795e4d85b5d9ceeca43f5fef	offer received	504	f19421c1d4aa40978ebb69ca19b0e20d	5	[web, email, mobile, social]	5	5	bogo	F	75	20170509	100000.0
4	78afa995795e4d85b5d9ceeca43f5fef	offer completed	510	f19421c1d4aa40978ebb69ca19b0e20d	5	[web, email, mobile, social]	5	5	bogo	F	75	20170509	100000.0
5	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	582	f19421c1d4aa40978ebb69ca19b0e20d	5	[web, email, mobile, social]	5	5	bogo	F	75	20170509	100000.0
6	78afa995795e4d85b5d9ceeca43f5fef	offer received	408	ae264e3637204a6fb9bb56bc8210ddfd	10	[email, mobile, social]	10	7	bogo	F	75	20170509	100000.0
7	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	408	ae264e3637204a6fb9bb56bc8210ddfd	10	[email, mobile, social]	10	7	bogo	F	75	20170509	100000.0
8	78afa995795e4d85b5d9ceeca43f5fef	offer completed	510	ae264e3637204a6fb9bb56bc8210ddfd	10	[email, mobile, social]	10	7	bogo	F	75	20170509	100000.0
9	78afa995795e4d85b5d9ceeca43f5fef	offer received	168	5a8bc65990b245e5a138643cd4eb9837	0	[email, mobile, social]	0	3	informational	F	75	20170509	100000.0
10	78afa995795e4d85b5d9ceeca43f5fef	offer viewed	216	5a8bc65990b245e5a138643cd4eb9837	0	[email, mobile, social]	0	3	informational	F	75	20170509	100000.0



event	viewed_to_received	completed_to_viewed	completed_to_received
offer_type			
bogo	0.830488	0.692290	0.574939
discount	0.692383	0.930848	0.644503
informational	0.703754	NaN	NaN

The data tells us that customers are extremely receptive to “bogo” offers if they see the offer having a conversion rate of 93.08%.

Perform feature engineering, final dataframe for building machine learning model:

	customer id	event	time	offer id	reward	difficulty	duration	age	income	offer_type_bogo	...	offer_type_informational	gender_F	gender_M	gender_O	email	mobile	social	web	month_member
0	0	1	0.000000	0	5	0.25	0.571429	75	0.777778	1	...	0	1	0	0	1	1	0	1	9
1	0	2	0.008403	0	5	0.25	0.571429	75	0.777778	1	...	0	1	0	0	1	1	0	1	9
2	0	3	0.184874	0	5	0.25	0.571429	75	0.777778	1	...	0	1	0	0	1	1	0	1	9
3	0	1	0.705882	1	5	0.25	0.285714	75	0.777778	1	...	0	1	0	0	1	1	1	1	9
4	0	3	0.714286	1	5	0.25	0.285714	75	0.777778	1	...	0	1	0	0	1	1	1	1	9
5	0	2	0.815126	1	5	0.25	0.285714	75	0.777778	1	...	0	1	0	0	1	1	1	1	9
6	0	1	0.571429	2	10	0.50	0.571429	75	0.777778	1	...	0	1	0	0	1	1	1	0	9
7	0	2	0.571429	2	10	0.50	0.571429	75	0.777778	1	...	0	1	0	0	1	1	1	0	9
8	0	3	0.714286	2	10	0.50	0.571429	75	0.777778	1	...	0	1	0	0	1	1	1	0	9
9	0	1	0.235294	3	0	0.00	0.000000	75	0.777778	0	...	1	1	0	0	1	1	1	0	9

Then we proceed to split the data into training dataframe and test dataframe to train and test the machine learning models.

K-Nearest Neighbor algorithm is used as the benchmark model.

Two other models were also trained, using the Random Forest algorithm and Decision Tree algorithm.

Looking at the F1 scores - A score of 1 indicates perfect precision and recall, while a score of 0 indicates that the model has no predictive power, Decision Tree algorithm performs the best overall as seen below:

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	51.970824	30.943498
1	RandomForestClassifier	93.948595	44.469093
2	DecisionTreeClassifier	93.949715	63.169305

Thus, proceed to tune the hyperparameters of the Decision Tree algorithm, to get a better model.

	Refined DecisionTreeClassifier Model	train F1 score	test F1 score
0	DecisionTreeClassifier	78.751176	74.004235

Evaluating the models further

Benchmark model, K-Nearest Neighbor algorithm

```
KNeighborsClassifier
--------------------
Train Report
              precision    recall  f1-score   support

           1       0.52      0.83      0.64     39812
           2       0.51      0.36      0.42     29875
           3       0.59      0.13      0.22     19565

    accuracy                           0.52     89252
   macro avg       0.54      0.44      0.43     89252
weighted avg       0.53      0.52      0.47     89252

Test Report
              precision    recall  f1-score   support

           1       0.38      0.58      0.46     26666
           2       0.18      0.14      0.15     19969
           3       0.09      0.03      0.04     12867

    accuracy                           0.31     59502
   macro avg       0.21      0.25      0.22     59502
weighted avg       0.25      0.31      0.27     59502

--------------------
```

Random Forest algorithm model

```
RandomForestClassifier
--------------------
Train Report
              precision    recall  f1-score   support

           1       0.92      0.97      0.94     39812
           2       0.95      0.89      0.92     29875
           3       0.97      0.95      0.96     19565

    accuracy                           0.94     89252
   macro avg       0.95      0.94      0.94     89252
weighted avg       0.94      0.94      0.94     89252

Test Report
              precision    recall  f1-score   support

           1       0.59      0.67      0.63     26666
           2       0.29      0.27      0.28     19969
           3       0.30      0.25      0.27     12867

    accuracy                           0.44     59502
   macro avg       0.39      0.40      0.39     59502
weighted avg       0.43      0.44      0.43     59502

--------------------
```

Decision Tree algorithm model

```
DecisionTreeClassifier
--------------------
Train Report
              precision    recall  f1-score   support

           1       0.90      1.00      0.95     39812
           2       0.97      0.87      0.92     29875
           3       1.00      0.92      0.96     19565

    accuracy                           0.94     89252
   macro avg       0.96      0.93      0.94     89252
weighted avg       0.94      0.94      0.94     89252

Test Report
              precision    recall  f1-score   support

           1       0.81      0.81      0.81     26666
           2       0.50      0.50      0.50     19969
           3       0.47      0.47      0.47     12867

    accuracy                           0.63     59502
   macro avg       0.59      0.59      0.59     59502
weighted avg       0.63      0.63      0.63     59502

--------------------
```

Tuned model, Decision Tree algorithm

```
DecisionTreeClassifier
--------------------
Train Report
              precision    recall  f1-score   support

           1       0.83      1.00      0.91     39812
           2       0.76      0.61      0.68     29875
           3       0.70      0.63      0.66     19565

    accuracy                           0.79     89252
   macro avg       0.76      0.75      0.75     89252
weighted avg       0.78      0.79      0.78     89252

Test Report
              precision    recall  f1-score   support

           1       0.83      0.99      0.90     26666
           2       0.68      0.54      0.60     19969
           3       0.59      0.54      0.56     12867

    accuracy                           0.74     59502
   macro avg       0.70      0.69      0.69     59502
weighted avg       0.73      0.74      0.73     59502

--------------------
```

Recap:

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	51.970824	30.943498
1	RandomForestClassifier	93.948595	44.469093
2	DecisionTreeClassifier	93.949715	63.169305
3	Refined DecisionTreeClassifier Model	78.751176	74.004235

Conclusion:

The proposed solution aims to analyze the customer behavior on the Starbucks rewards mobile app and identify which customers respond best to offers. The final model (Refined DecisionTreeClassifier Model) serves decently well as a base model to be further developed by collecting more data for further training, testing other algorithm which might be developed in future, etc. With further development, the machine learning model will eventually be able to accurately predict customer behavior.

In conclusion, this project has the potential to provide Starbucks with valuable insights into customer behavior on their mobile app, enabling them to tailor their marketing campaigns to better reach and engage their customers. By using machine learning techniques, we can predict customer behavior accurately and improve the overall customer experience.