Machine Learning Engineer Nanodegree

Starbuck's Capstone Challenge

Matteo Giuliani
October 15th, 2024

I. Definition

Project Overview

Starbucks, like many companies, wants to make sure that their customers are aware of and use the special offers and promotions they send out. These offers could include discounts on coffee or snacks, or buy-one-get-one-free deals. The main challenge is to figure out how to make sure that the right offers are sent to the right customers—essentially, understanding what kinds of promotions different customers like and are likely to respond to.

Problem Domain:

Starbucks sends out different types of offers to customers through its mobile app, such as discounts, BOGO (buy-one-get-one-free) deals, or even just information about a new product. But not all customers are interested in every offer. Some people might be more inclined to respond to a 20% discount, while others might be more interested in trying a new product for free. The goal is to use data to identify which types of offers are most effective for which customers, and when the best time is to send them.

Project Origin:

This project comes from a real-world problem that Starbucks faces as it tries to improve customer engagement and satisfaction. By analyzing data about customer behavior and the effectiveness of different offers, we can help Starbucks better understand its customers and send out promotions that they are more likely to appreciate and use. This means a better experience for customers and more successful marketing efforts for Starbucks.

Data Sets and Input Data:

The project uses data that includes:

- **Customer Profiles (profiles.json)**: Information about customers, such as their age, income, and when they became members.
- **Offers Data (portfolio.json)**: Details about the different offers that were sent out, including the type of offer and its duration.
- **Transaction Data (transcript.json)**: Records of purchases made by customers, showing whether they responded to offers.

The challenge is to analyze this data and create a model that predicts which offers each customer is likely to respond to, allowing Starbucks to better target its promotions and improve customer satisfaction. The ultimate goal is to optimize how offers are sent out to improve both customer experience and sales.

Problem Statement

The primary challenge is to determine which types of promotional offers are most effective for different customers, based on their preferences and behaviors. Starbucks needs a way to match each offer type—such as discounts, BOGO (buy-one-get-one-free) deals, or new product informational to the customers who are most likely to respond positively. This problem arises from the need to improve the effectiveness of marketing efforts, which in turn could enhance customer satisfaction and increase revenue.

The goal is to build a predictive model that can analyze customer data and predict the likelihood that a customer will respond to a particular offer. This model will allow Starbucks to make data-driven decisions when sending out offers, ensuring that customers receive promotions that are relevant to their interests and habits.

To solve this problem, the following strategy will be employed:

1. **Data Exploration**: Investigate the structure and quality of the dataset, identifying key features and understanding how customer demographics, offers, and transactions are related.
2. **Data Preprocessing**: Clean and preprocess the data, handling any missing values, reaname columns or split aggregated data.
3. **Exploratory Data Analysis (EDA)**: Analyze the relationships between customer demographics, purchase behaviors, and offer responses to identify trends and insights that can inform model building.
4. **Model Selection and Training**: Train two machine learning models: a Random Forest and a Decision Tree. These models will be designed to predict the likelihood of a customer responding to an offer.
5. **Model Evaluation**: Compare the performance of the Random Forest and Decision Tree models against a benchmark K-Neighbors Classifier. The primary evaluation metric will be the F1 score, which balances precision and recall, providing a measure of a model's effectiveness in identifying positive responses to offers.

Anticipated Solution

The intended solution is a predictive model that identifies which offers are most suitable for each customer segment. By sending personalized offers, Starbucks can increase the engagement rate of their promotions and ensure that customers receive offers they are more likely to use.

This solution is expected to improve marketing efficiency, reducing the costs associated with sending irrelevant offers and increasing customer satisfaction. Customers benefit from receiving promotions that match their preferences, while Starbucks benefits from higher conversion rates and increased sales. Additionally, the analysis could provide deeper insights into customer behavior, helping Starbucks make more informed decisions regarding future promotions and marketing strategies.

Metrics

For this project, I will build two models using **RandomForestClassifier** and **DecisionTreeClassifier**, and compare their **F1 score** against a **KNeighborsClassifier** benchmark.

Metric Selection

- **F1 Score**: The primary metric for comparison, as it balances **precision** and **recall**. This is crucial for the Starbucks Challenge, where both false positives (predicting a response that doesn't occur) and false negatives (missing a responder) matter.

Model Comparison

- **RandomForestClassifier**: Uses multiple decision trees for robust predictions and reduces overfitting.
- **DecisionTreeClassifier**: A simpler model that is easier to interpret but more prone to overfitting.
- **KNeighborsClassifier**: Serves as a benchmark model, offering a straightforward comparison point for more complex models.

Each model's F1 score will be compared to see if RandomForest or DecisionTree significantly outperforms the benchmark, helping select the best model for predicting customer responses to Starbucks offers.

II. Analysis

Data Exploration

The dataset consists of three distinct files:

- **portfolio.json** - Contains details about various offers, including their IDs and specific attributes like type and duration.
- **profile.json** - Includes demographic details for each customer.
- **transcript.json** - Tracks all records of interactions, including transactions, receipt of offers, views, and completions.

Below is a description of the structure and details for each variable found in the files:

portfolio.json

- **id** (string) - Unique identifier for each offer.
- **offer_type** (string) - Describes the nature of the offer, such as "Buy One Get One," discounts, or informational.
- **difficulty** (int) - The minimum expenditure required to qualify for the offer.
- **reward** (int) - The incentive given upon successful completion of the offer.
- **duration** (int) - Validity period of the offer, measured in days.
- **channels** (list of strings) - Communication methods used for the offer.

profile.json

- **age** (int) - The customer's age.
- **became_member_on** (int) - The registration date when the customer joined the app.
- **gender** (str) - Indicates the customer's gender (note: some entries include 'O' for non-binary or other).
- **id** (str) - Unique identifier for each customer.
- **income** (float) - The annual earnings of the customer.

transcript.json

- **event** (str) - Describes the type of interaction (e.g., transaction, receipt of an offer, viewing of an offer).
- **person** (str) - Identifies the customer associated with each interaction.
- **time** (int) - Indicates the time in hours since the beginning of the testing period, starting at hour zero.
- **value** (dict of strings) - Contains either a transaction amount or an offer ID, depending on the interaction type.

Exploratory Visualization

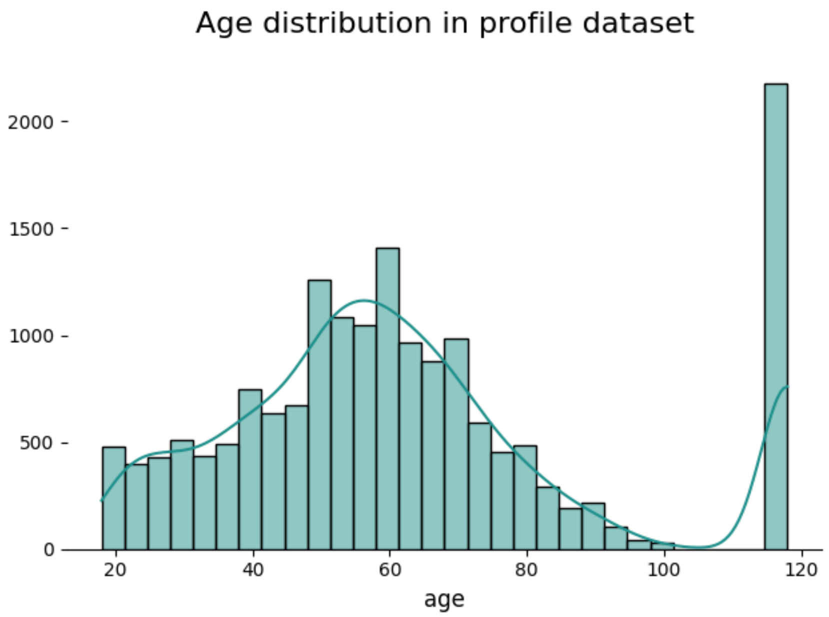
Profile dataset visualization

```
the profile dataset has 17000 rows and 5 columns
```

	gender	age	id	became_member_on	income
count	14825	17000.000000	17000	1.700000e+04	14825.000000
unique	3	NaN	17000	NaN	NaN
top	M	NaN	68be06ca386d4c31939f3a4f0e3dd783	NaN	NaN
freq	8484	NaN	1	NaN	NaN
mean	NaN	62.531412	NaN	2.016703e+07	65404.991568
std	NaN	26.738580	NaN	1.167750e+04	21598.299410
min	NaN	18.000000	NaN	2.013073e+07	30000.000000
25%	NaN	45.000000	NaN	2.016053e+07	49000.000000

	gender	age	id	became_member_on	income
50%	NaN	58.000000	NaN	2.017080e+07	64000.000000
75%	NaN	73.000000	NaN	2.017123e+07	80000.000000
max	NaN	118.000000	NaN	2.018073e+07	120000.000000

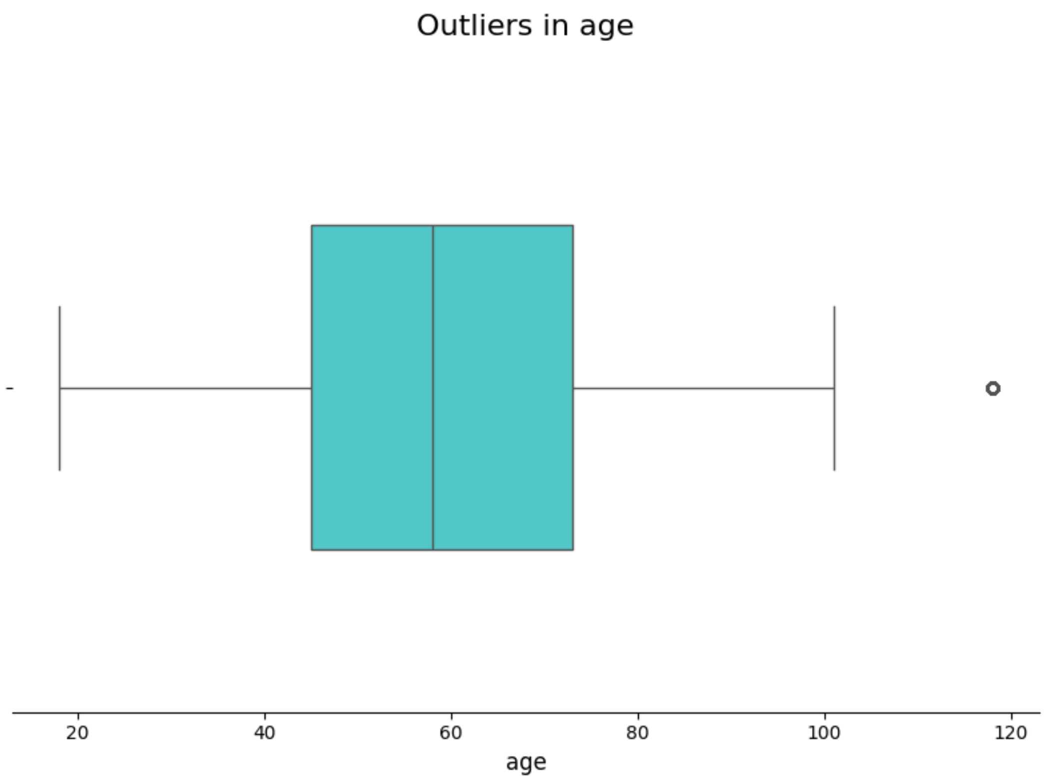
	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN



	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN

	gender	age	id	became_member_on	income
...	...	...	...	...	...
16980	None	118	5c686d09ca4d475a8f750f2ba07e0440	20160901	NaN
16982	None	118	d9ca82f550ac4ee58b6299cf1e5c824a	20160415	NaN
16989	None	118	ca45ee1883624304bac1e4c8a114f045	20180305	NaN
16991	None	118	a9a20fa8b5504360beb4e7c8712f8306	20160116	NaN
16994	None	118	c02b10e8752c4d8e9b73f918558531f7	20151211	NaN

A large number of rows have an age value of 118. Upon examining these rows, it is evident that when the age is 118, the gender and income fields are null. I will use this information to clean the dataset later.



Individuals who are older than 80 years seem to exhibit lower engagement with the app, suggesting they might also have lower beverage consumption. Therefore, I classify this age group as outliers in the dataset.

Transcript dataset visualization

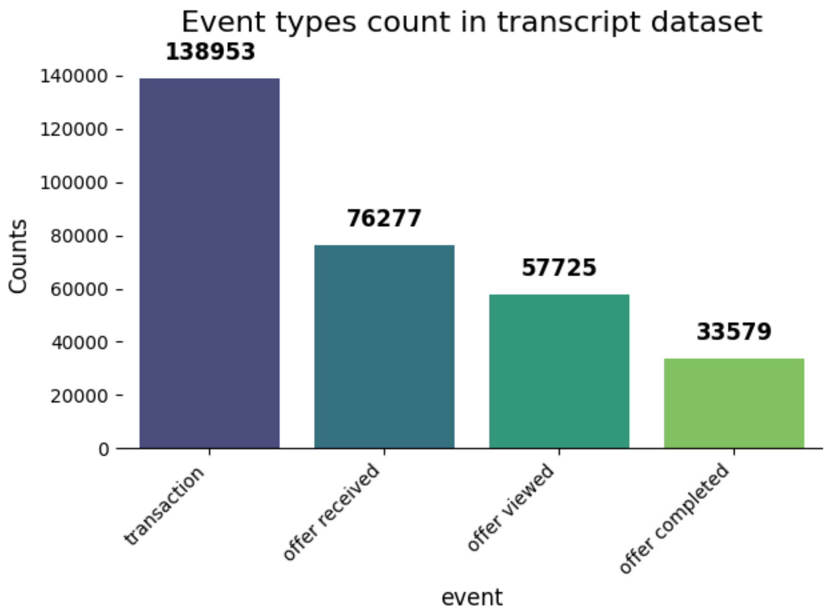
```
the transcript dataset has 306534 rows and 4 columns
```

	person	event	value	time
count	306534	306534	306534	306534.000000
unique	17000	4	5121	NaN
top	94de646f7b6041228ca7dec82adb97d2	transaction	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}	NaN
freq	51	138953	14983	NaN
mean	NaN	NaN	NaN	366.382940
std	NaN	NaN	NaN	200.326314
min	NaN	NaN	NaN	0.000000
25%	NaN	NaN	NaN	186.000000
50%	NaN	NaN	NaN	408.000000
75%	NaN	NaN	NaN	528.000000
max	NaN	NaN	NaN	714.000000

	person	event	value	time

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
2	e2127556f4f64592b11af22de27a7932	offer received	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0

```
We have 4 different possibility in the transcript dataset for the value field:
'offer id': 134002 occurrencies
'amount': 138953 occurrencies
'offer_id, reward': 33579 occurrencies
```



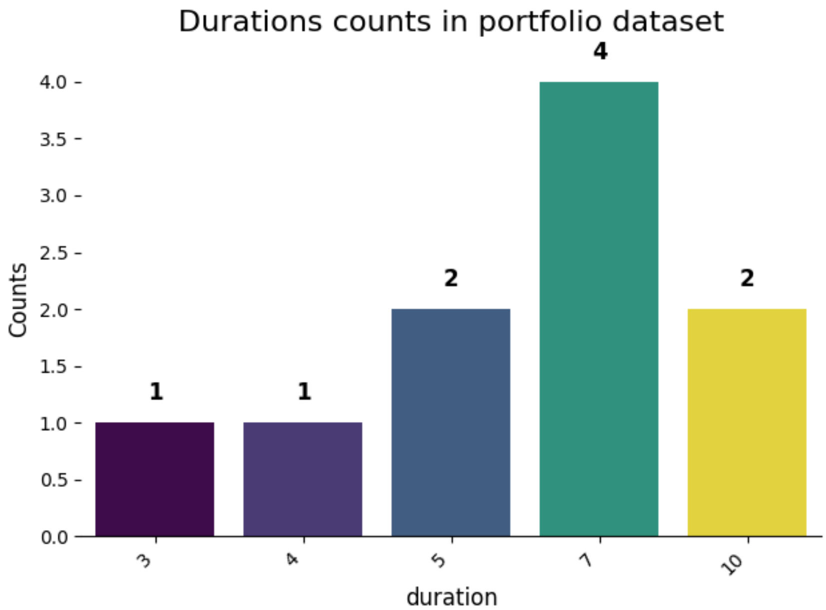
Portfolio dataset visualization

```
the portfolio dataset has 10 rows and 6 columns
```

	reward	channels	difficulty	duration	offer_type	id
count	10.000000	10	10.000000	10.000000	10	10
unique	NaN	4	NaN	NaN	3	10
top	NaN	[web, email, mobile, social]	NaN	NaN	bogo	ae264e3637204a6fb9bb56bc8210ddfd
freq	NaN	4	NaN	NaN	4	1
mean	4.200000	NaN	7.700000	6.500000	NaN	NaN
std	3.583915	NaN	5.831905	2.321398	NaN	NaN
min	0.000000	NaN	0.000000	3.000000	NaN	NaN
25%	2.000000	NaN	5.000000	5.000000	NaN	NaN
50%	4.000000	NaN	8.500000	7.000000	NaN	NaN
75%	5.000000	NaN	10.000000	7.000000	NaN	NaN
max	10.000000	NaN	20.000000	10.000000	NaN	NaN

	reward	channels	difficulty	duration	offer_type	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

	reward	channels	difficulty	duration	offer_type	id
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7



Algorithms and Techniques

To predict customer responses to promotional offers, we will use **Random Forest** and **Decision Tree** algorithms.

1. **Random Forest**:
    - **Description**: An ensemble method that builds multiple decision trees and combines their results.
    - **Justification**:
        - **Robustness**: Reduces the risk of overfitting, which is helpful with complex customer data.
        - **Feature Importance**: Identifies which customer traits most influence response to offers.
2. **Decision Tree**:
    - **Description**: A model that splits data into branches based on feature values.
    - **Justification**:
        - **Interpretability**: Easy to understand how decisions are made based on customer data.
        - **Non-linear Relationships**: Captures complex patterns in customer responses.

Data Handling

- **Data Exploration**: We will assess the dataset to identify key features related to customer demographics and offer responses.
- **Data Preprocessing**: This step includes cleaning the data, handling missing values, and encoding categorical variables.

- **Exploratory Data Analysis (EDA)**: We will analyze trends and relationships in the data to inform model building.

Benchmark

We will use a **K-Neighbors Classifier (KNN)** as a benchmark for evaluating our models.

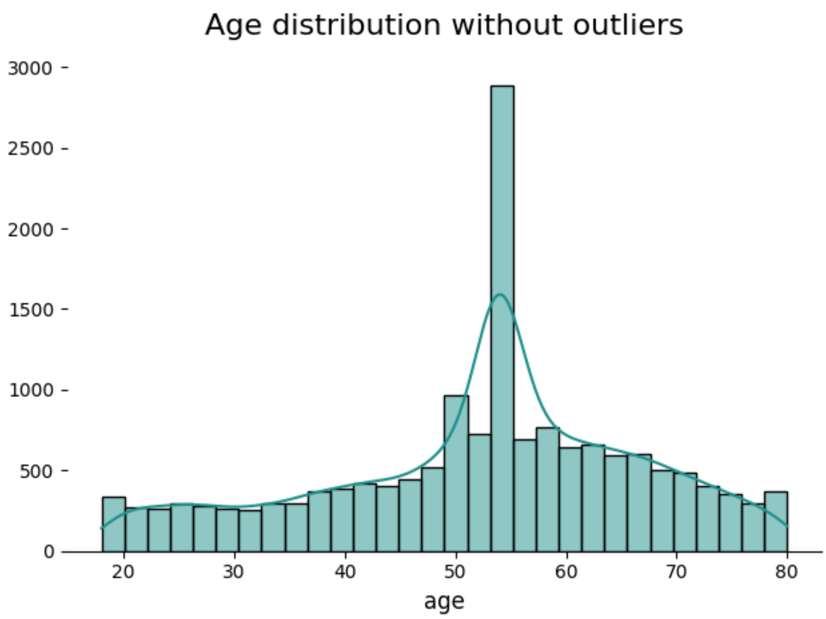
1. **Benchmark Definition**:
    - The KNN model will serve as a baseline, with performance measured using the **F1 score**, which balances precision and recall.
2. **Rationale for Benchmark**:
    - KNN is a simple yet effective algorithm that provides a good starting point for classification tasks.
    - The F1 score is particularly useful for our problem, as it addresses potential imbalances between positive and negative customer responses.
3. **Performance Measurement**:
    - We expect both the Random Forest and Decision Tree models to achieve higher F1 scores than KNN, indicating better predictive accuracy for customer responses.

III. Methodology

Data Preprocessing

Cleaning profile dataset

1. To retain as much data as possible, I imputed missing values using the mean for age and income, and the mode for gender.
2. Individuals over the age of 80 are outliers as emerged from the exploratory phase and excluded from the dataset.



3. I've categorized ages into groups for better clarity during Exploratory Data Analysis (EDA):
    - Under 20
    - 20 - 45
    - 46 - 60
    - 61 - 80
4. I've renamed columns(`id` -> `customer_id`, `income` -> `customer_income`) to improve readability and facilitate merging of dataframes.

	gender	customer_id	became_member_on	customer_income	age_group
0	M	68be06ca386d4c31939f3a4f0e3dd783	20170212	65404.991568	46-60
1	F	0610b486422d4921ae7d2bf64640c50b	20170715	112000.000000	46-60
2	M	38fe809add3b4fcf9315a9694bb96ff5	20180712	65404.991568	46-60
3	F	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.000000	61-80
4	M	a03223e636434f42ac4c3df47e8bac43	20170804	65404.991568	46-60
5	M	e2127556f4f64592b11af22de27a7932	20180426	70000.000000	61-80
6	M	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	65404.991568	46-60
7	M	68617ca6246f4fbc85e91a2a49552598	20171002	65404.991568	46-60

	gender	customer_id	became_member_on	customer_income	age_group
8	M	389bc3fa690240e798340f5a15918d5c	20180209	53000.000000	61-80
9	M	8974fc5686fe429db53ddde067b88302	20161122	65404.991568	46-60

Clening transcript dataset

1. I've expanded the nested keys in the 'value' column into separate new columns. The "value" column contains dictionaries, with each key from these dictionaries separated into its own column.
   - `offer id` or `offer_id` will go in the `offer_id` columns.
   - `amount` will go in the `money_spent` columns.
   - `reward` will go in the `money_gained` columns.

	person	event	time	offer_id	money_gained	money_spent
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	0.0
1	a03223e636434f42ac4c3df47e8bac43	offer received	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	0.0
2	e2127556f4f64592b11af22de27a7932	offer received	0	2906b810c7d4411798c6938adc9daaa5	0	0.0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	0	fafdcd668e3743c1bb461111dcafc2a4	0	0.0
4	68617ca6246f4fbc85e91a2a49552598	offer received	0	4d5c57ea9a6940dd891ad53e9dbe8da0	0	0.0

2. I've renamed the column (`person` -> `customer_id`) to enhance readability and simplify the process of merging dataframes.

	customer_id	event	time	offer_id	money_gained	money_spent
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	0.0
1	a03223e636434f42ac4c3df47e8bac43	offer received	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	0.0
2	e2127556f4f64592b11af22de27a7932	offer received	0	2906b810c7d4411798c6938adc9daaa5	0	0.0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	0	fafdcd668e3743c1bb461111dcafc2a4	0	0.0
4	68617ca6246f4fbc85e91a2a49552598	offer received	0	4d5c57ea9a6940dd891ad53e9dbe8da0	0	0.0

Cleaning portfolio dataset

	reward	channels	difficulty	duration	offer_type	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

1. I've renamed columns (`difficulty` -> `offer_difficulty`, `id` -> `offer_id`, `duration` -> `offer_duration`, `reward` -> `offer_reward`) to enhance readability and simplify the process of merging dataframes.

	offer_reward	channels	offer_difficulty	offer_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

Merging the dataframes

I've merged the dataframes joining transcript and portfolio on `offer_id` and then joining the resulting dataframe with profile on `customer_id`.

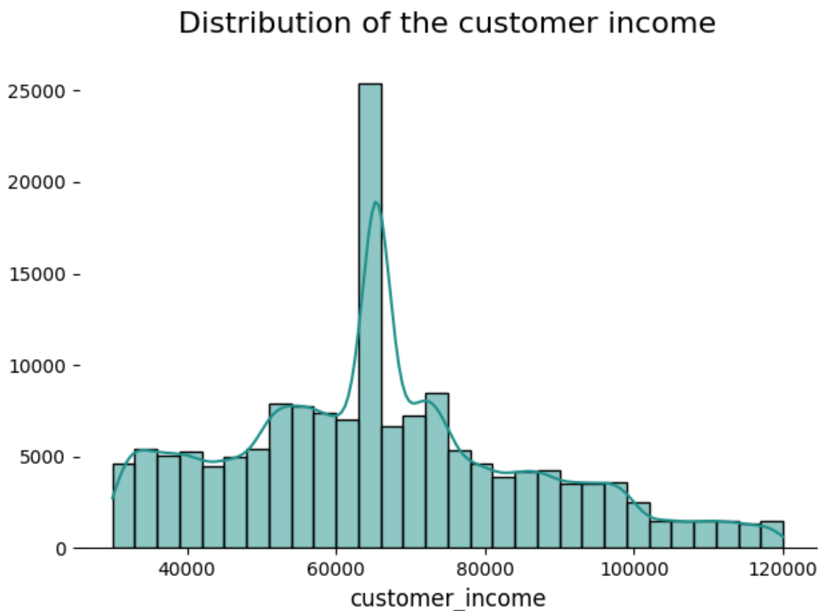
	offer_reward	channels	offer_difficulty	offer_duration	offer_type	offer_id	customer_id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	4b0da7e80e5945209a1fdddfe813

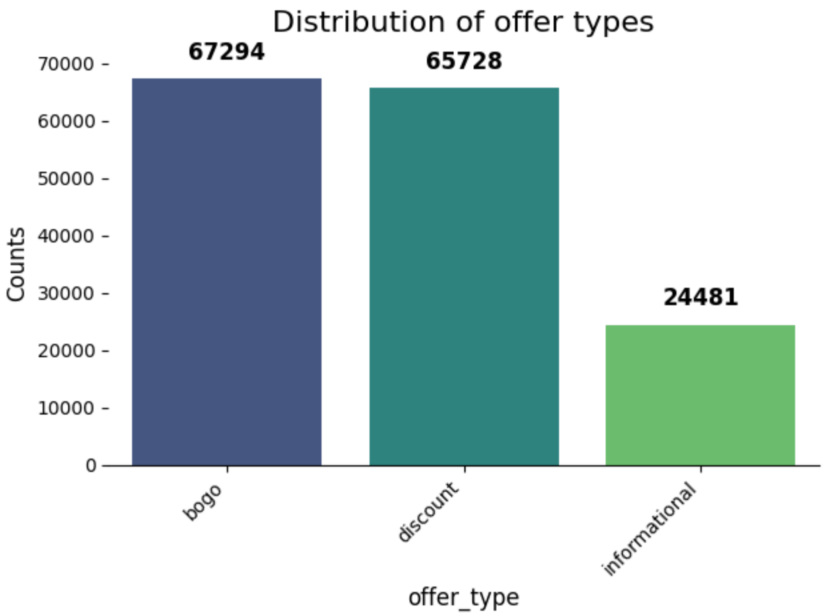
	offer_reward	channels	offer_difficulty	offer_duration	offer_type	offer_id	customer_id
1	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1e9420836d554513ab90eba9855
2	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	02c083884c7d45b39cc68e1314f
3	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	676506bad68e4161b9bbaffeb03
4	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	fe8264108d5b4f198453bbb1fa7c
5	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	39dbcf43e24d41f4bbf0f134157e0
6	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	3f244f4dea654688ace14acb4f02
7	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	92e07c49ee7448fca6e48df0c96e
8	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	f8aedd0cbea0419c806842b4265
9	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	8a4bc602e4424ab6b16f0b907f2f

Implementation

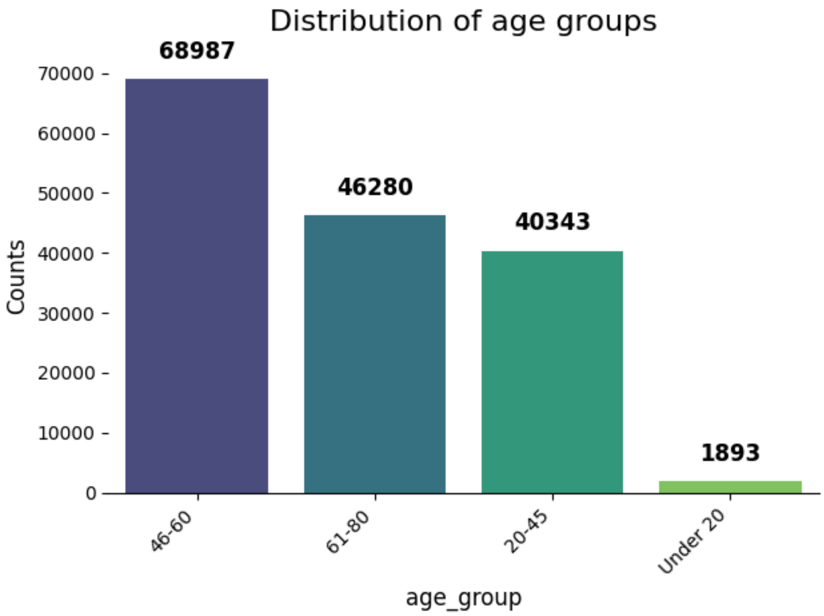
Exploratory data analysis (EDA)

```
The average income of customers is: 65924.49109976534
```

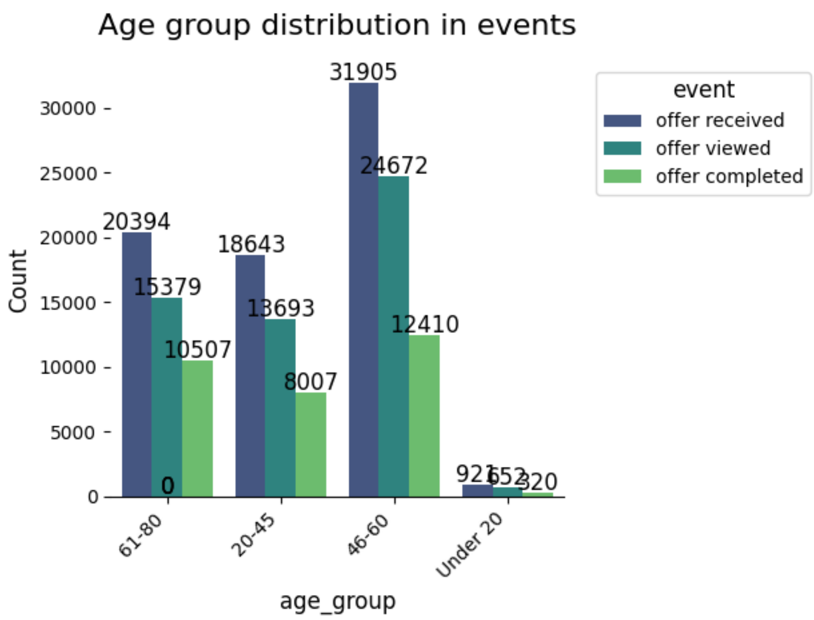


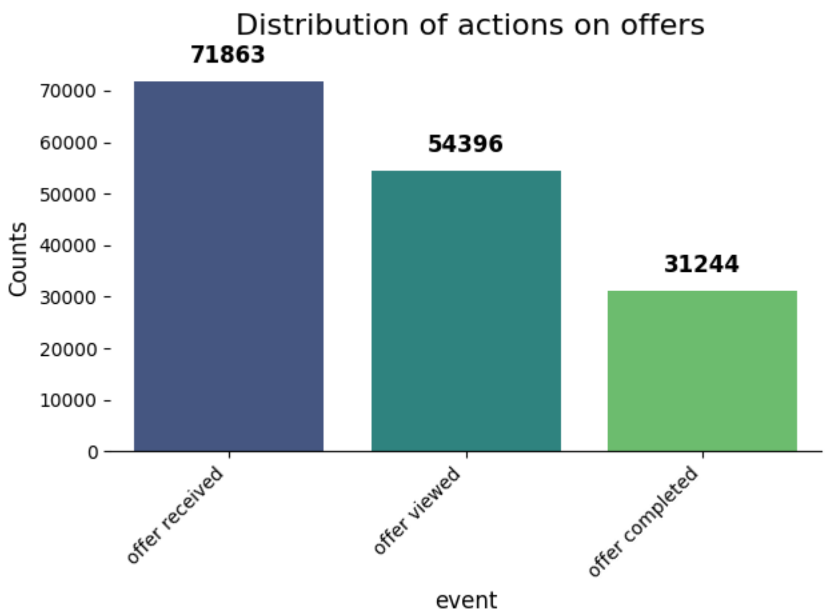


The majority of the offers are `BOGO` and `Discount`.

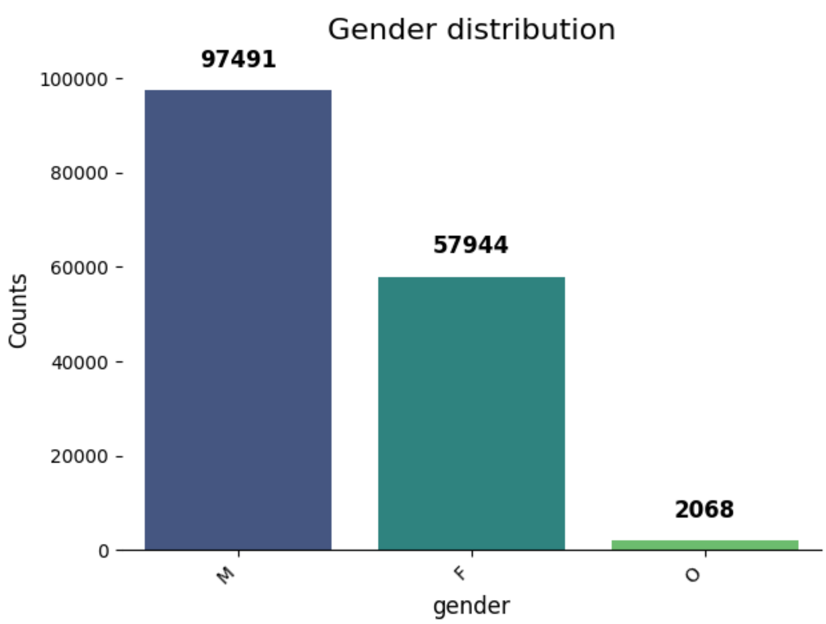


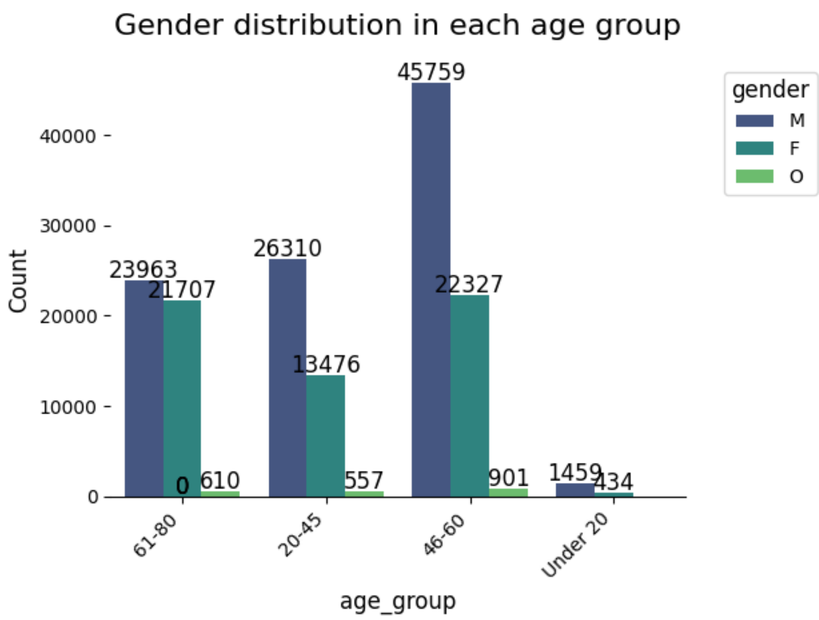
Contrary to common expectations, the Starbucks app is most popular among users aged 46-60, with those aged 61-80 coming in second. Surprisingly, the younger demographic of 20-45, who are often assumed to be the primary app users, do not dominate usage in this instance.



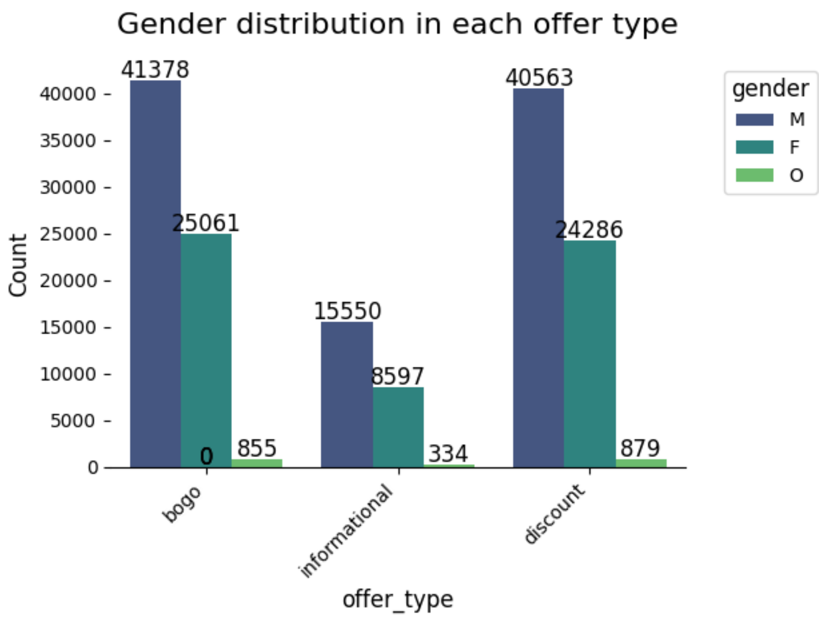


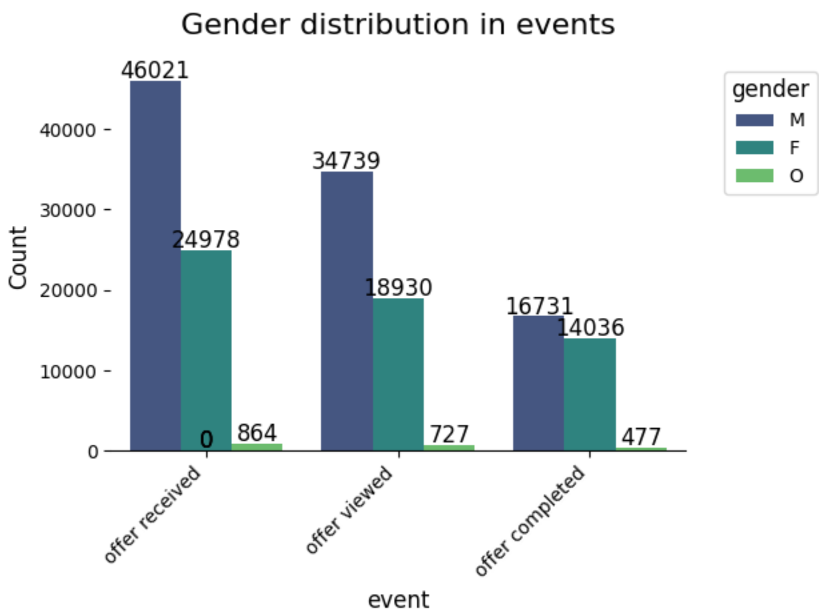
This suggests that the majority of customers disregard the offer entirely, not even taking a moment to review it. Additionally, more customers simply view and dismiss the offer compared to those who proceed to complete it.

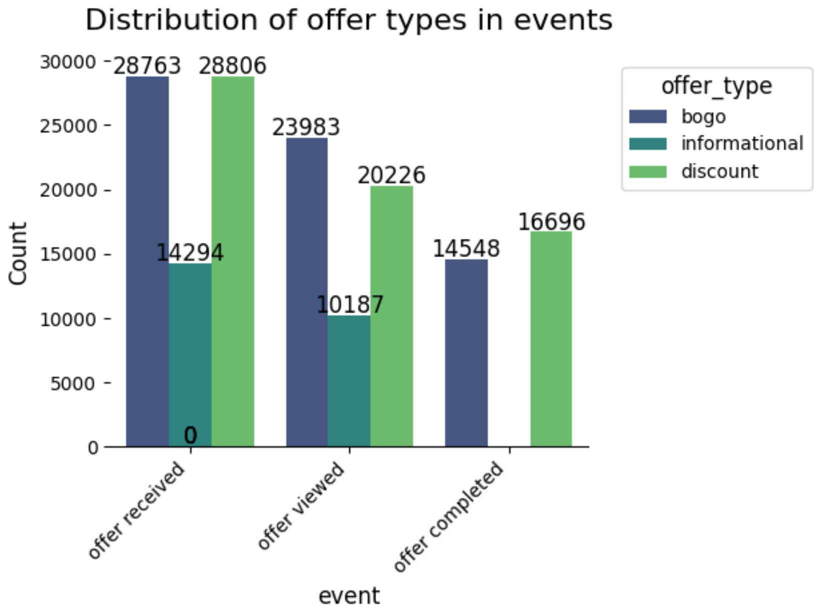




In every age group, there are more male customers than female customers







Overall, the majority of people tend to take advantage of the discount offer.

- Observations

Males account for 62.7% of the data and tend to use the Starbucks app more frequently than females. Notably, both males and females in the 46-60 age group are the heaviest users of the app. Customers show a stronger preference for discount offers. However, there is a lower number of customers who actually complete offers compared to those who simply view and ignore them.

Training data preparation

- I've performed one-hot-encoding of columns with categorical values (`gender`, `offer_type`, `age_group`)

	offer_reward	channels	offer_difficulty	offer_duration	offer_id	customer_id	eve
0	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	4b0da7e80e5945209a1fdddfe813dbe0	offe rece
1	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	1e9420836d554513ab90eba98552d0a9	offe rece
2	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	02c083884c7d45b39cc68e1314fec56c	offe rece
3	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	676506bad68e4161b9bbaffeb039626b	offe rece
4	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	fe8264108d5b4f198453bbb1fa7ca6c9	offe rece

- I've encoded the `event` values with numerical values.

	offer_reward	channels	offer_difficulty	offer_duration	offer_id	customer_id	eve
0	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	4b0da7e80e5945209a1fdddfe813dbe0	1
1	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	1e9420836d554513ab90eba98552d0a9	1
2	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	02c083884c7d45b39cc68e1314fec56c	1

	offer_reward	channels	offer_difficulty	offer_duration	offer_id	customer_id	eve
3	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	676506bad68e4161b9bbaffeb039626b	1
4	10	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	fe8264108d5b4f198453bbb1fa7ca6c9	1

- I've converted the `offer_id` and `customer_id` columns into numerical format.

	offer_reward	channels	offer_difficulty	offer_duration	offer_id	customer_id	event	time	money_gained	money_spent	...	gen
0	10	[email, mobile, social]	10	7	0	0	1	0	0	0.0	...	Fals
1	10	[email, mobile, social]	10	7	0	1	1	0	0	0.0	...	Fals
2	10	[email, mobile, social]	10	7	0	2	1	0	0	0.0	...	True
3	10	[email, mobile, social]	10	7	0	3	1	0	0	0.0	...	Fals
4	10	[email, mobile, social]	10	7	0	4	1	0	0	0.0	...	True

- I've removed the `became_member_on` column and create separate columns for the month and year.

	offer_reward	channels	offer_difficulty	offer_duration	offer_id	customer_id	event	time	money_gained	money_spent	...	gen
0	10	[email, mobile, social]	10	7	0	0	1	0	0	0.0	...	Fals
1	10	[email, mobile, social]	10	7	0	1	1	0	0	0.0	...	Fals
2	10	[email, mobile, social]	10	7	0	2	1	0	0	0.0	...	Fals
3	10	[email, mobile, social]	10	7	0	3	1	0	0	0.0	...	Fals
4	10	[email, mobile, social]	10	7	0	4	1	0	0	0.0	...	Fals

- I've dropped the `channel` colum

	offer_reward	offer_difficulty	offer_duration	offer_id	customer_id	event	time	money_gained	money_spent	customer_income
0	10	10	7	0	0	1	0	0	0.0	100000.0
1	10	10	7	0	1	1	0	0	0.0	70000.0
2	10	10	7	0	2	1	0	0	0.0	30000.0
3	10	10	7	0	3	1	0	0	0.0	92000.0
4	10	10	7	0	4	1	0	0	0.0	93000.0

- I've scaled and normalized the numerical values and removed `channel` column.

	offer_reward	offer_difficulty	offer_duration	offer_id	customer_id	event	time	money_gained	money_spent	customer_income
0	1.0	0.5	0.571429	0	0	1	0.0	0.0	0.0	0.777778
1	1.0	0.5	0.571429	0	1	1	0.0	0.0	0.0	0.444444
2	1.0	0.5	0.571429	0	2	1	0.0	0.0	0.0	0.000000
3	1.0	0.5	0.571429	0	3	1	0.0	0.0	0.0	0.688889
4	1.0	0.5	0.571429	0	4	1	0.0	0.0	0.0	0.700000
5	1.0	0.5	0.571429	0	5	1	0.0	0.0	0.0	0.377778
6	1.0	0.5	0.571429	0	6	1	0.0	0.0	0.0	0.455556
7	1.0	0.5	0.571429	0	7	1	0.0	0.0	0.0	0.311111
8	1.0	0.5	0.571429	0	8	1	0.0	0.0	0.0	0.466667
9	1.0	0.5	0.571429	0	9	1	0.0	0.0	0.0	0.011111

Split training and test data

```
Training set contains: 94501 rows
Testing set contains: 63002 rows
```

Training

The K-Nearest Neighbors algorithm is used to establish the benchmark, and the model's performance is evaluated using the F1 score metric. I've trained a `RandomForestClassifier` and `DecisionTreeClassifier` with the data and these are the initial results compared to the Benchmark.

	Model	train F1 score	test F1 score
0	KNeighborsClassifier	54.287256	33.337037
1	RandomForestClassifier	95.443434	70.699343
2	DecisionTreeClassifier	95.443434	84.933812

Refinement

Intermediate Steps and Improvements

After assessing the initial results, I noticed that both the **RandomForestClassifier** and **DecisionTreeClassifier** performed better than the benchmark **KNeighborsClassifier**. However, I aimed to improve their performance further through hyperparameter tuning and model optimization.

- Random Forest Classifier Tuning I adjusted parameters such as the number of trees (`n_estimators`) and the maximum depth of the trees (`max_depth`) to enhance the model's performance. I tested various combinations using randomized search.
- Decision Tree Classifier Tuning For the **DecisionTreeClassifier**, I also performed hyperparameter tuning by adjusting the maximum depth and the minimum samples required to split a node.

These are the results after the tuning:

	Model	train F1 score	test F1 score
0	KNeighborsClassifier	54.287256	33.337037
1	RandomForestClassifier	92.096380	74.767468
2	DecisionTreeClassifier	92.454048	91.920891

IV. Results

Model Evaluation and Validation

The F1 score will be utilized as the primary metric for evaluating the effectiveness of the approach and identifying the model that yields the most favorable results. This score can be understood as the weighted average of precision and recall. Specifically, the balanced F-score, commonly known as the F1 score, represents the harmonic mean of precision and recall. Its values range from 0 to 100, with a score of 100 indicating optimal performance and 0 representing the worst outcome.

	Model	train F1 score	test F1 score
0	KNeighborsClassifier	54.287256	33.337037
1	RandomForestClassifier	92.096380	74.767468
2	DecisionTreeClassifier	92.454048	91.920891

Justification

The validation set was used to evaluate the performance of different machine learning models in predicting customer responses to marketing offers. The KNeighborsClassifier served as the baseline for comparison, achieving a test F1 score of **33.34**. Both the RandomForestClassifier and DecisionTreeClassifier significantly outperformed this baseline.

Among the models tested, the **DecisionTreeClassifier** achieved the highest test F1 score of **91.92**, indicating its effectiveness in classifying customer responses to promotional offers. The **RandomForestClassifier** also demonstrated strong performance, with a test F1 score of **74.77**. Both models exhibit a considerable improvement over the baseline, showcasing their capability in effectively predicting customer engagement.

Since the primary goal is to predict customer responses to marketing offers, an extremely high F1 score isn’t strictly necessary. The performance metrics suggest that both the DecisionTreeClassifier and RandomForestClassifier are robust enough for practical application in this context. Consequently, their scores are deemed satisfactory for our needs.

In summary, while the baseline model has its limitations, both the RandomForestClassifier and DecisionTreeClassifier substantially enhance predictive accuracy. These models are particularly well-suited for the Starbucks Capstone Challenge, indicating their potential to effectively predict customer engagement with offers and provide valuable insights for marketing strategies.