STARBUCKS OFFER OPTIMIZATION

PROJECT REPORT

1. Background

Starbucks is a coffee company and coffeehouse chain that serves hot and cold drinks, various kinds of coffee and tea. Once every few days, Starbucks sends out an offer to users of their mobile app as a way to stimulate customer spending. Starbucks is looking to optimise their offering strategy so that the right offer is sent to the right customer.

With millions of customers and various types of offers, it is impossible to allocate personnel to manually decide the offering for each customer. Each person in the simulation has some hidden traits that influence their purchasing patterns and are associated with their observable traits. People produce various events, including receiving offers, opening offers, and making purchases.

Therefore, it is necessary to build an automated decision process that allocate the right offer to the right customer.

2. Dataset

Starbucks has provided a dataset that contains simulated data that mimics customer behaviour on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be an advertisement for a drink or an actual offer. Some users might not receive any offer during certain weeks. The data is contained in three files, which will be explained as follows:

2.1. Portfolio data

This contains offer ids and meta data about 10 available offers (duration, type, etc.) sent during 30-day test period

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational. In details:
  - Buy-one-get-one (BOGO): A user needs to spend a certain amount to get a reward equal to that threshold amount.
  - Discount: A user gains a reward equal to a fraction of the amount spent
  - Informational: There is no reward, but neither is there a requisite amount that the user is expected to spend.
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward (in USD) given for completing an offer
- duration (int) - time for offer to be open, in days
- channels (list of strings) web, email, mobile, social

2.2. Customer profile data

This contains demographic data for each customer, aging from 18 to 118, with approximately 15,000 customers, in which 6000 females, 8000 males, and 200 other, income ranging from 30,000 USD per annum to 120,000 USD per annum.

- age (int) - age of the customer, missing value encoded as 118
- became_member_on (int) - date when customer created an app account, format YYYYMMDD
- gender (str) - gender of the customer (M, F, O, or null)
- id (str) - customer id
- income (float) - customer's income

2.3. Transaction data

This contains records for approximately 300,000 events such as offers received, offers viewed, and offers completed. This shows user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase.

This transactional data also has a record for each offer that a user receives as well as a record for when a user views the offer. There are also records for when a user completes an offer. It's also important to know that a user can receive an offer, never actually view the offer, and still complete the offer. For example, a user might receive the "buy 10 dollars get 2 dollars off offer", but the user never opens the offer during the 10 day validity period. The customer spends 15 dollars during those ten days. There will be an offer completion record in the data set; however, the customer was not influenced by the offer because the customer never viewed the offer.

- event (str) - record description (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

3. Methods

3.1. Solution statement overview

The proposed solution has 2 parts, outlined as follows:

Predict customer spending

This part aims to predict how customer would spend with and without the influence of offer, based on their profiles (age, gender, income, time, etc.).

In this part, information on offer, customer profile and historical spending are combined into 1 data table with the following information:

Table 1 - Combined dataset

Source	Information	Note
portfolio.json	offer type	Label encoded or one-hot encoded
	channels	Label encoded or one-hot encoded
profile.json	age	Might need to be binned depending on model type

Source	Information	Note
	income	Might need to be binned depending on model type
	id	Might be included/removed depending on data availability
transcript.json	Spending	Spending during the period when the offer is valid, which is also the target variable

Then, relevant features will be engineered, depending on the findings from the exploratory data analysis part. Data is then split into training and validation set, with 70% in the training set and 20% in the validation set. A machine learning algorithm is then chosen to learn to predict customer spending given an offer.

Create an offer policy to maximise customer spending

Since the above step generate a simulation of customer spending if given each offer, the decision rule will simply choose the offer that maximise the increased spending.

The sections below outline the detailed steps to achieve the above objectives.

3.2. Data transformation

The original datasets were provided in json format, which were transformed into csv to facilitate data analysis.

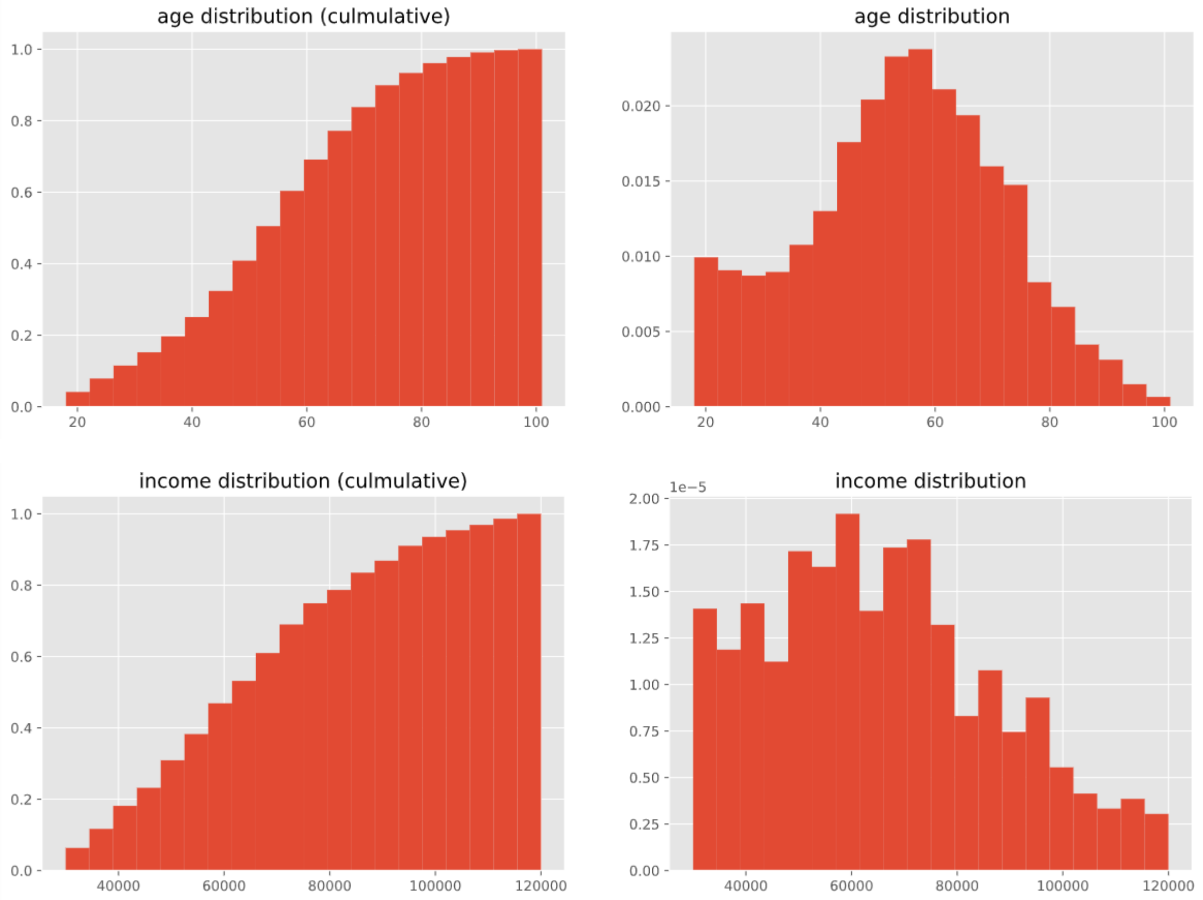
3.3. Exploratory data analysis

This part aims to apply statistical and visualisation techniques to obtain an overview understanding of the data and informs later stages of the analysis. The findings are presented as follows

3.3.1. Customer profiles

From the dataset, 80% of customers are above 40 years old, as suggested below, and the average income is approximately $70,000. Given the median weekly income of a full-time salary worker in the US is $957 per week or ~ $50,000 per annum according to the US Department of Labour here, it can be concluded that more than 80% of Starbucks' customers earn more than average.

Figure 1 - Customer age and income distribution



Analysing customer profile by gender suggested that more than 60% of customers are male, 40% were female and a small percentage are “Other”, which consists of 212 customers.

3.3.2. Portfolio

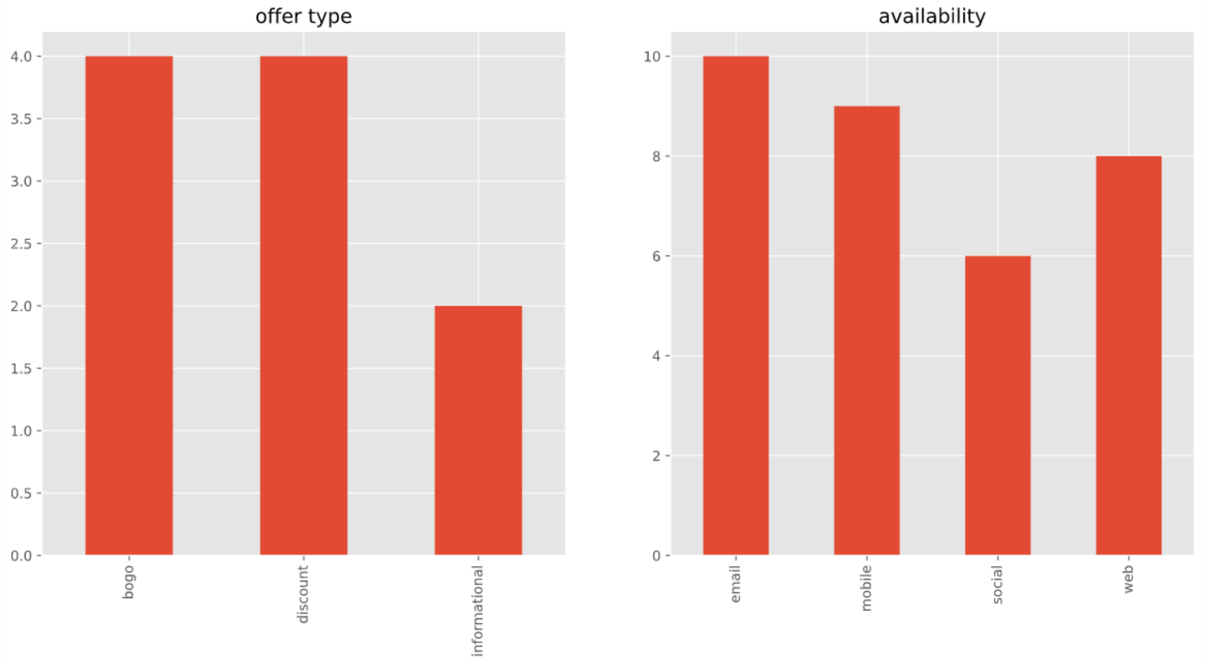
The table below suggests that on average, the reward is $4.2, and customers need to spend $7.7 to claim the offer. This approximates to a 54% discount. Offers last for 156 hours or 6.5 days on average.

Table 2 - Summary statistics for the Portfolio dataset

	reward	difficulty	duration
count	10	10	10
mean	4.2	7.7	156
std	3.58392	5.83191	55.7136
min	0	0	72
max	10	20	240

In the portfolio, there are 4 discount offers, 2 informational offers and 4 buy-one-get-one offers. All of them are available through email, 9 through mobile, 6 through social media and 8 through the website.

Figure 2 - Offer type and availability



3.3.3. Customer spending

<u>*Overview*</u>

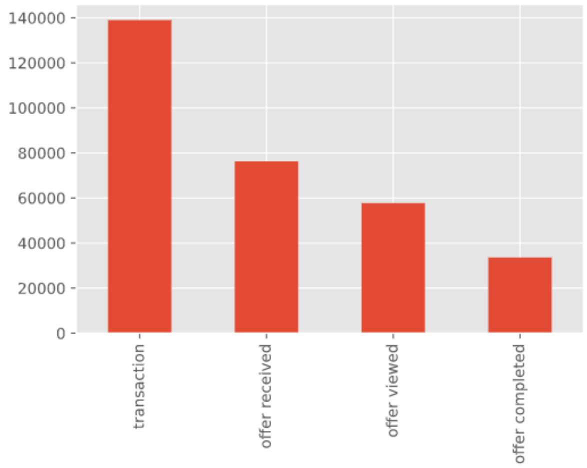
Overall, the average spending of each transaction is $30, and average reward is $3. Therefore, the net amount received by Starbucks was approximately $27.

Table 3 - Summary statistics for customer spending

	amount	reward
count	138953	33579
mean	12.77736	4.904137
std	30.25053	2.886647
min	0.05	2
max	1062.28	10

Analysing the distribution of events in the transaction log shows that there were 138953 purchases, 76277 offer receipts 57725 offer views, and 33579 offer completions. Therefore, the offer completion rate is 33579 / 76277 ~ 44%, offer view rate is 57725/76277 ~ 75%.

Figure 3 - Count by event type in transaction log



Analysing the value distribution of the orders shows that most of the orders were below $50, and there were some orders at $1000. It is not plausible to tell if these orders are outliers or not, because there is a possibility that some customers consistently spend more than a few hundred dollars on drinks. Analysis on median order value distribution confirms the above possibility: There were 3 customers who consistently spend more than $100 on drinks.

Figure 4 - Distribution of order values

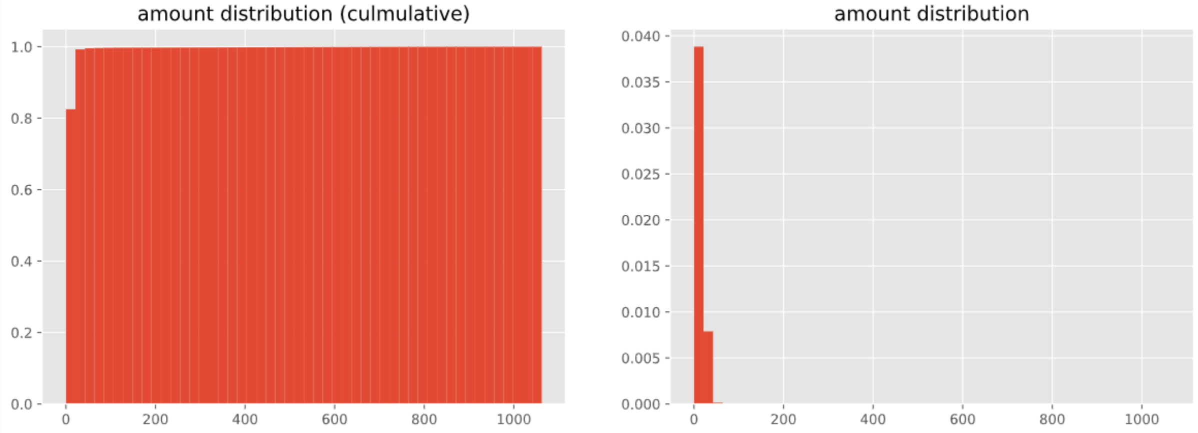
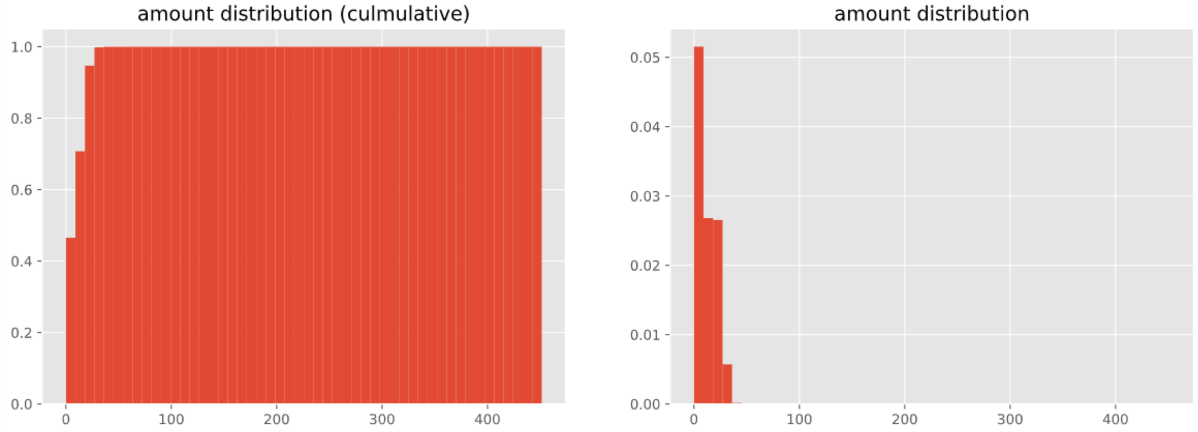


Figure 5 - Distribution of median customers' order value

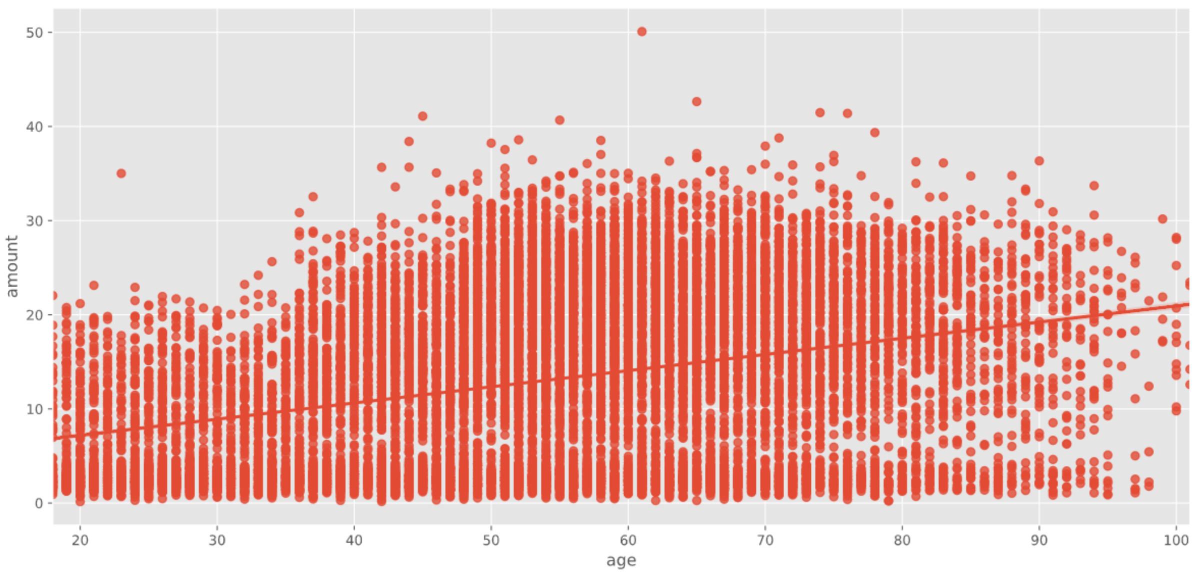


As a result, these customers were considered as outliers and removed from the analysis.

Order value vs. age

The plot suggests that on average, the more senior the customer is, the more they spend on an order. This is true because as age correlates with career progression in general, thus more money to spend on drinks.

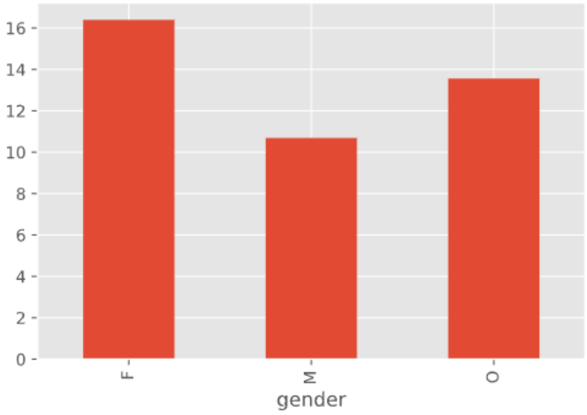
Figure 6 - Median order value by age



Order value vs. gender

The plot suggests that on average, female spend the most on an order.

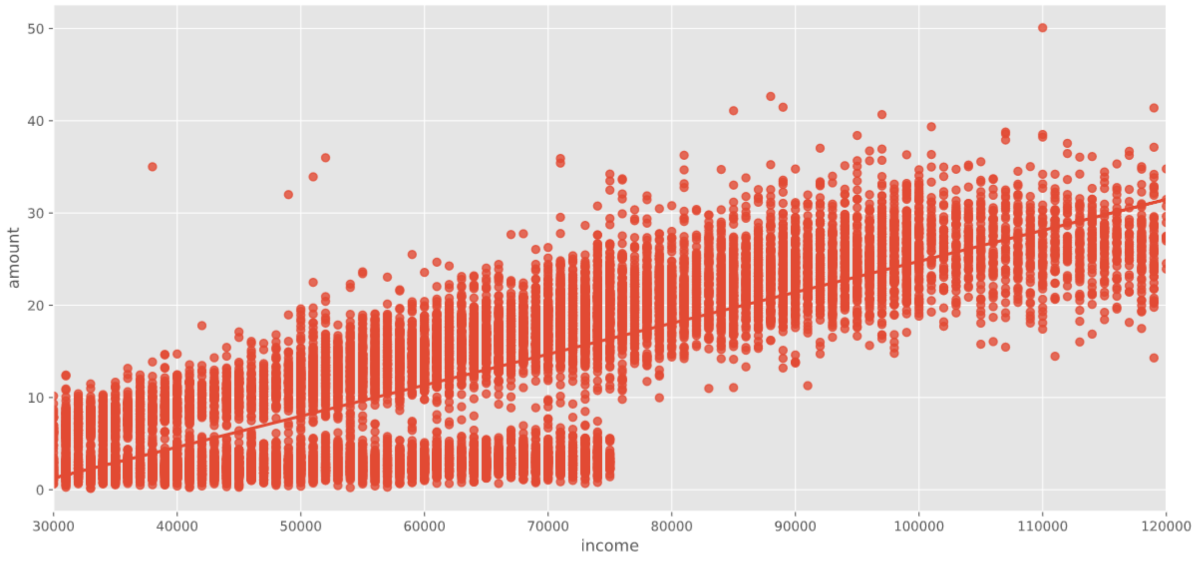
Figure 7 - Median order value by gender



Order value vs. income

Since the objective is to predict how a customer would spend with or without an order, it's important to see if it is actually predictable by looking at factors that might drive spending. In this case, income could be a candidate. The figure below shows the correlation between income and median order spending by customer

Figure 8 - Income vs. median order spending by customer



The plot suggests that on average, the higher the income, the higher the order value. However, the bottom-left corner suggests that a significant number of customers only spend less than $5 on an order regardless of their income. This means that income alone, although correlates well with average order spending, is not the sole factor determining order value. Therefore, analysing other factors is necessary.

In addition, this also shows that typically, orders are below $50. Therefore, orders valued at thousands or hundred dollars are considered outliers, thus removed from the dataset.

3.4. Data cleansing and pre-processing

3.4.1. Customer profile data

There are approximately 2,200 missing values in gender, income, and age. Interestingly, those with missing gender also have missing income. Analysing customer spending in section 3.3 suggested that age and income are the 2 most promising predictors of spending. Therefore, predicting spending for these customers might be inaccurate. As a result, these 2,200 customers will be removed from the analysis.

The “became_member_on” column contains the timestamp when the user registered for membership. In general, it is unlikely that it will be related to customer spending behaviour. A potential direction could be to analyse their spending pattern as they progress in their membership life. This requires a reference to measure how long have they been member, which is not available in the dataset. Nevertheless, analysing this could be too big to fit in the scope of this project. Therefore, this column was removed from the dataset.

3.4.2. Portfolio data

The table below shows the first 2 lines of the portfolio dataset. The channels through which the offers are available is encoded as a list. To numerically represent this, the channels are binary-encoded into separate columns. The “offer_type” column is also binary-encoded, with the first resulting column being dropped to avoid the dummy variable trap.

Table 4 - First 2 rows of the raw portfolio dataset

reward	channels	difficulty	duration	offer_type	id
10	['email', 'mobile', 'social']	10	7	bogo	ae…
10	['web', 'email', 'mobile', 'social']	10	5	bogo	4d…

The first 2 rows of the resulting table are as follows:

Table 5 - First 2 rows of the transformed portfolio dataset

offer	reward	difficulty	duration	email	mobile	social	web	discount	informational
ae…	10	10	168	1	1	1	0	0	0
4d...	10	10	120	1	1	1	1	0	0

3.4.3. Transaction log data

Removing outliers

As identified in section 3.3.3, 3 customers and transactions identified as outliers were removed from the analysis.

Matching transactions with offers

The transaction log records purchases and transactions separately. If a purchase gets discounted by an offer, a separate record with event type of “offer completed” is logged with the same timestamp as the transaction’s (highlighted in blue).

Table 6 - Transaction log for customer 78afa995795e4d85b5d9ceeca43f5fef

amount	reward	event	time	offer
		offer received	408	ae264e3637204a6fb9bb56bc8210ddfd
		offer viewed	408	ae264e3637204a6fb9bb56bc8210ddfd
		offer received	504	f19421c1d4aa40978ebb69ca19b0e20d
21.72		transaction	510	
	10	offer completed	510	ae264e3637204a6fb9bb56bc8210ddfd
	5	offer completed	510	f19421c1d4aa40978ebb69ca19b0e20d
26.56		transaction	534	
		offer viewed	582	f19421c1d4aa40978ebb69ca19b0e20d

Since our objective is to predict customer spending with and without an offer, it is necessary to fill the “offer” column of the “transaction” event with the offer that was completed such as below:

Table 7 - Desire output for offer-transaction matching

amount	reward	event	time	offer
21.72		transaction	510	ae264e3637204a6fb9bb56bc8210ddfd

In addition, it can be seen in Table 6, highlighted in yellow, that the offer *f19421c1d4aa40978ebb69ca19b0e20d* was viewed after the purchase, yet it still showed up as completed in the transaction log, which is incorrect. To address this issue, the following algorithm, descried as pseudo-code below, was developed to track the status of the offers (received, viewed, expired, etc.) overtime and match the transaction with the corresponding offer:

```
for transaction in customer_transcript:
    for all_offers in offer tracking list:
        Decay offer validity

    if event_type = 'offer received':
      add offer to offer tracking list
    else if event_type == 'offer viewed':
      add offer to viewed list
    else if event_type == 'transaction':
      record transaction with list of viewed offers
    else if event_type == 'offer completed':
      if last_event == 'transaction':
        Remove the last transaction as it does not contain
          the offer it should have
        Record transaction with list of viewed offers
```

The above steps result in a 141,209 rows x 3 columns table matching the transactions with their used offers. The first 3 rows are as follows:

Table 8 – Resultant table of matching transactions and offers

person	amount	offer
78afa995795e4d85b5d9ceeca43f5fef	19.89	9b98b8c7a33c4b65b9aebfe6a799e6d9
78afa995795e4d85b5d9ceeca43f5fef	17.78	no_offer
78afa995795e4d85b5d9ceeca43f5fef	19.67	5a8bc65990b245e5a138643cd4eb9837

3.5. Model selection

Since this is a regression task, linear regression was selected as the baseline model. This algorithm also outputs a list of coefficients, which is interpretable and can be used to validate the findings from the exploratory data analysis section.

LightGBM was selected as the main machine learning model to predict customer spending. The reason being LightGBM is fast, scalable and proven to be a powerful learner.

3.6. Feature engineering

3.6.1. Feature extraction and transformation

The feature extraction and transformation steps are summarised as follow:

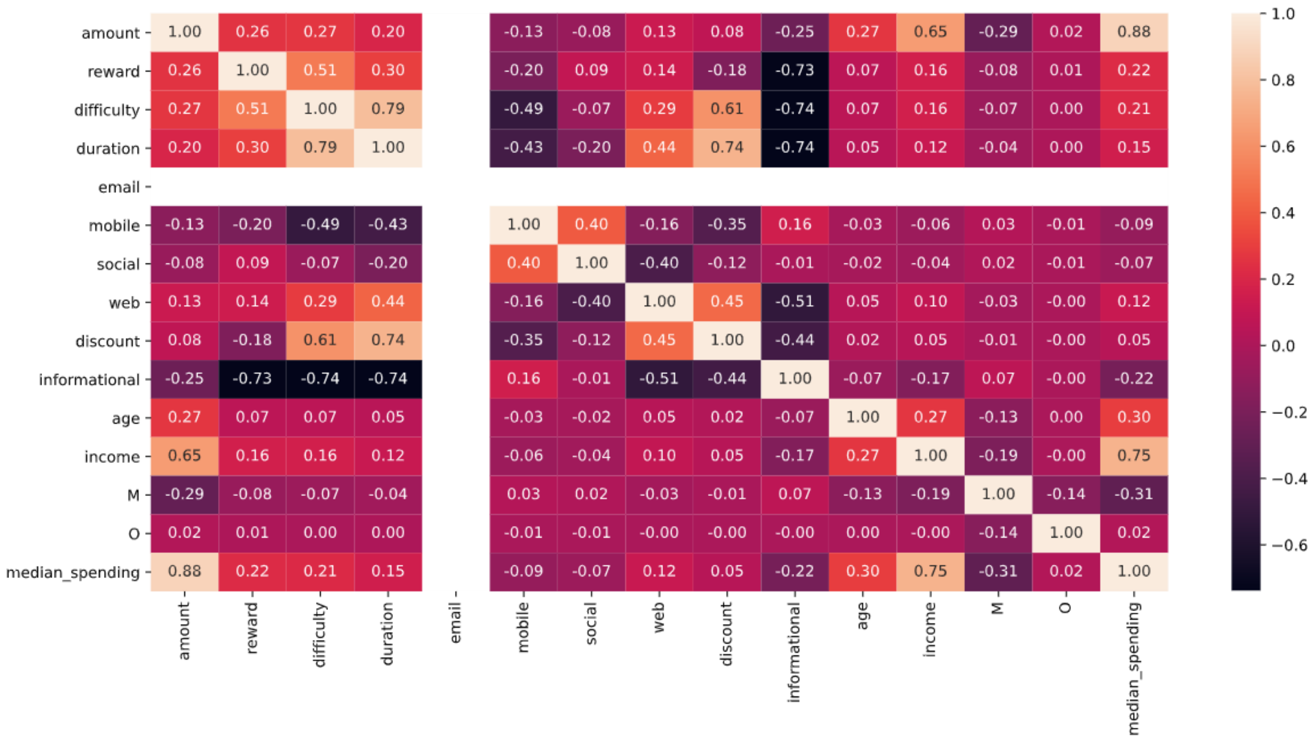
One-hot encode gender: Customer genders are currently encoded as M, F, and O in the profile dataset, they need to be encoded as 1 and 0 so it can be consumed by the machine learning model

Add median spending: To capture the typical amount that someone would spend on a drink

Merge all dataset: Information on customer and offer attributes are merged to the transformed transaction log resulted from section 3.4.3 to capture all the possible information on a transaction.

Remove the "email" column: Since all offers are available through emails, the "email" column will only contain the value *1*, which does not add any information to the feature set, as suggested in the figure below:

Figure 9 - Pairwise feature correlation plot



3.6.2. Feature engineering for LightGBM

The "person" column containing customer ids is currently encoded as strings. It was then numerically encoded by flagging each customer with a unique number.

A more sophisticated approach could be to create an embedding matrix to represent this feature, with each customer ids being represented as a vector. This is a powerful approach as it allows customer with similar spending pattern having similar vectors. However, given the scope of this project, this approach was not pursuit.

3.6.3. Feature engineering for Linear Regression

Figure 9 suggested that there is a strong correlation between median spending, amount, and income. This is not desirable for linear models as it can make the model unstable and decrease performance on unseen data. Therefore, the "median_spending" feature was dropped.

To represent customer ids, there are 3 approaches:

- Label encode: This is not ideal for linear models, which assume a linear relationship between the variable and the target.
- One hot encode: This adds a sparse n_transactions x n_customers matrix to the feature set, which is again not desirable
- Embedding: Decided not to pursuit per the above analysis

Therefore, the "person" column representing customer ids was dropped from the feature set for linear regression.

The input data was then normalised to 0 mean and unit variance.

3.7. Model training

3.7.1. LightGBM

Hyperparameters for LightGBM was selected based on a greedy approach that iterates over a set of predefined parameter space. The following set yielded the best result:

Table 9 - Hyperparameters set for LightGBM

hyperparameter	value
boosting_type	gbdt
objective	rmse
max_depth	2
num_leaves	5
learning_rate	0.1
bagging_fraction	0.7
bagging_freq	1
verbose	1
num_boost_round	200

3.7.2. Linear Regression

Linear regression was trained using the default configuration specified in the scikit-learn's documentation.

3.8. Simulate customer spending

In this part, the most performing model in predicting customer spending was then used to predict customer spending under the influence of the 10 offers, and no offer available.

Then, based on the spending prediction, the offer / no offer that yields the highest spending was selected for each customer.

3.9. Validation

3.9.1. Predicting customer spending

Splitting the data into training and test set

The dataset was randomly split into a training and validation sets, with 70% and 30% of the records going to the training and validation set respectively.

The training set is used to train the machine learning model to learn to predict customer spending on an order, while the validation set is used to validate the predicting performance of the model on new data.

Metrics

Since this is a regression task with no clear preference for over prediction or under prediction, Root mean squared error was selected as the metric to evaluate the model performance.

3.9.2. Simulating customer spending

Businesses may seek to optimise for maximum income (i.e. customer spending). Therefore, spending increased by allocating an offer to customer (in USD) was selected as the evaluating metric.

Spending increased is calculated as follows:

spending_increased = **spending_with_offer - spending_without_offer**

In which: spending_with_offer, spending_without_offer is the amount that a customer would spend if they receive or did not receive an offer.

4. Results

4.1. Predicting customer spending

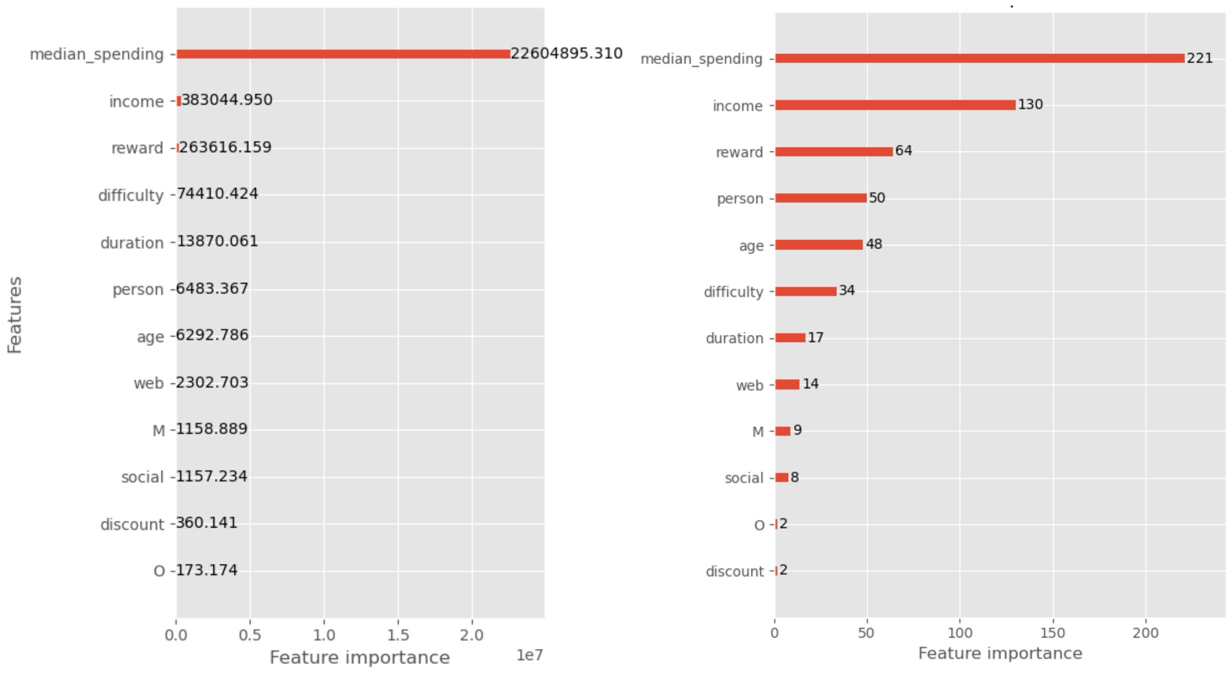
The table below summarises the results obtained from model training:

Table 10 - Error on the training and validation set

Model	Training error (RMSE)	Testing error (RMSE)
LightGBM	3.25	3.23
Linear Regression	5.04	5.10

For LightGBM, the results suggested that on average, the model's prediction is off by $3, which is arguably reasonable given the average value of the offers were from $30 to $40, which approximates to a 10% error.

Figure 10 - Feature importance by gain (left) and split (right) for LightGBM



In terms of feature importance, median_spending appears to be the most important feature as it topped the feature importance chart. “Income”, “reward”, and “person”, although are less importance in terms of gain, they helped fine tune the model as suggested by the plot on the right.

Linear regression did worse than LightGBM, which is expected as it's a baseline model and 2 features were dropped. However, some insights can be noted:

- Spending positively correlates with reward and difficulty, which is resonable. The higher the reward is, the more inclined the customers to spend more, especially for percentage discounts. Also, the higher the difficulty, the higher the minimum spending to meet the offer is.
- Spending increased with income, and women spends more money on drinks, which is consistent with our findings in the exploratory data analysis part.

4.2. Simulating customer spending

Simulating customer spending with and without offers suggested that the most optimal offers for 14,488 customers are as follow:

Table 11 - Summary of optimum offers for customers

Offer id	reward	channels	difficulty	duration	offer_type	count
ae264e3637…	10	Email, mobile, social	10	7	bogo	6628
0b1e1539f2…	5	Web, email	20	10	discount	7860

It appears that the most optimum offers were:

- Buy one get one, customer pay minimum 10 dollars and get a free drink with maximum value of 10 dollars (ae264e3637204a6fb9bb56bc8210ddfd)
- Pay 20, get 25% off (0b1e1539f2cc45b7b9fa7c272da2e1d7)

with a slight preference for the discount one. In addition, since “no_offer” did not show up in the optimum offer list, it can be seen that customers are stimulated by offers in general.

From the simulation results, by assigning these offers to customers, the average spending increase per offer is $4.25, which corresponds to a 105% increase. The results for the first 10 customers are as follows:

Table 12 - Simulated spending for the first 1 customers

person	no_offer_spending	optimum_offer_spending	spending_increased	pct_change
6813	23.231156	25.091666	1.86051	8.00
12815	18.295934	20.948339	2.652405	14.49
3167	11.541955	14.073435	2.53148	21.93
2611	3.556137	9.597052	6.040915	169.87
9721	12.472459	14.760702	2.288243	18.34
2796	19.55484	22.060166	2.505326	12.81
4265	29.074745	31.191126	2.116381	7.27
11066	19.481874	22.039774	2.5579	13.12
12077	10.953477	13.847586	2.894109	26.42
14033	15.856154	18.848653	2.9925	18.87

5. Conclusions

In this project, the following objectives were achieved:

- Predicting customer spending with and without offers: A linear regression (baseline) and a LightGBM model were trained to predict customer spending. LightGBM surpassed linear regression, predicting customer spending with an average error of $3 dollar. This is considered reasonable as the average spending was approximately $4.
- Choosing the optimum offer that maximise customer spending: Customer spending without offer and with each of the 10 offers were simulated, and a policy was developed to choose the best offer for each customer. This resulted in a $4.25 increase in spending, which corresponds to 105%. The analysis suggested that the most 2 efficient offers were buy one get one and discount.

Given the limited amount of time allocated to completing the project, it is certain that the analysis can be improved. Potential directions are as follow:

- Build a separate model to predict spending for customer who consistently spend more than 90% of the rest (who were removed from the analysis).
- Take into account membership lifetime as a factor to determine spending pattern
- Apply entity embedding to represent customer ids
- More sophisticated hyper parameter search for LightGBM
- Take into account multiple offers at once
- Impute missing information for the 2,200 customers who were removed from the dataset because of missing gender, age, and income.