Machine Learning Engineer Nanodegree

Capstone Project Report

by Arwa Alamoudi

23/12/2021

Introduction

In 2020, there were 32,660 Starbucks stores worldwide and the number has been slightly increased in 2021 despite the coronavirus pandemic [1]. Also, a study was conducted to understand the popularity of restaurants' loyalty applications. It found that Starbucks has the most regularly used loyalty rewards application amongst other major restaurants. This is due to the fact that Starbucks allows its customers to actively engage with the application and thus receive rewards [2][3].

With this active engagement, it would be very beneficial to leverage the data produced by Starbucks' digital customers to better understand their behavior and thus provide the needed services.

So, this project aims at understanding Starbucks' digital customers behavior through studying their reaction with the application offers. This will help Starbucks' team provide more customized offers that will keep these customers more attracted, thus satisfied.

The project is part of the Machine Learning Nanodegree program at Udacity. The data used in this capstone project is provided by Starbucks, it simulates how people make purchasing decisions and how those decisions are influenced by promotional offers.

Problem Statement

In this project, we want to know which groups of people are influenced by each type of offer. So, the goal is to better understand the customers hence send appropriate offers that increase Starbucks profits. We will focus on the customers who really get influenced by the offers. So, we send them appropriate offers. By influence we mean the customers who completely respond to the offers, that is, receive the offer, view it, process it, and complete it.

The following are the offers proposed by Starbucks in its reward application: BOGO (Buy One Get One Free), discount, and informational. This is a classification problem where the input is a type of customer and the output is an appropriate type of offer so Starbucks marketers can better understand their customers.

Metrics

Accuracy, Precision, Recall [4], F1, and Confusion Matrix is calculated to evaluate and compare between the built models.

Accuracy

It is the ratio of the number of correct predictions to the total number of input samples.

$$Accuracy = \frac{True\ positive + True\ negative}{Total}$$

Precision

It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}$$

Recall

It is the number of correct positive results divided by the number of all relevant samples**.**

$$Recall = \frac{True\ positive}{True\ positive + False\ negative}$$

F1 Score

It's used to measure the test's accuracy.

$$F1 = 2\ X\ \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Confusion Matrix

A metrics that summarizes the performance of a classification model. The metrics provide the following: true positive, true negative, false positive, false negative.

Analysis

There are three datasets in this project, and the goal is to explore each one individually before merging them. The datasets are:

1. **portfolio:** which contains information about the offers sent during the last 30 days.
2. **Profile:** which contains information about 17,000 customers.
3. **transcript:** which contains 306,534 offer reaction records.

portfolio

From initial exploration these are the observations:

- There are 10 rows and 6 columns.
- The portfolio dataset contains 6 columns:
    - reward (numeric): the money -in dollar- awarded for the amount spent.
    - channels (list): the channels used when sending the offer; web, email, mobile, social.
    - difficulty (numeric): the money required to be spent in order to receive the reward.
    - duration (numeric): time for an offer to be open, in days.
    - offer type (string): one of three types; Buy-One-Get-One "bogo", discount, informational.
    - id (string/hash): a unique id for each offer.
- The dataset contains no null values.

	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
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

Figure 1: portfolio Dataset

profile

From initial exploration these are the observations:

- There are 17,000 rows and 5 columns.
- The dataset contains 5 fields:
    - gender (categorical): customer gender; Male (M), Female (F), Other (O), or null.
    - age (numeric): customer age, missing value encoded as 118.
    - id (string/hash): a unique id for each customer.
    - became member on (date): customer joining date, written in format YYYYMMDD.
    - income (numeric): customer income.
- Gender, income, and age fields have 2,175 null values, all in the same rows.
- Customers' income ranges from 30,000 to 120,000 with an average of 65,404 and a standard deviation of 21,598.
- Customers' age ranges from 18 to 101 with an average of 54 and a standard deviation of 17.
- There are 8,484 males, 6,129 females, and 212 others.

	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
...	...	...	...	...	...
16995	F	45	6d5f3a774f3d4714ab0c092238f3a1d7	20180604	54000.0
16996	M	61	2cb4f97358b841b9a9773a7aa05a9d77	20180713	72000.0
16997	M	49	01d26f638c274aa0b965d24cefe3183f	20170126	73000.0
16998	F	83	9dc1421481194dcd9400aec7c9ae6366	20160307	50000.0
16999	F	62	e4052622e5ba45a8b96b59aba68cf068	20170722	82000.0

17000 rows × 5 columns

Figure 2: profile Dataset

The following are some visualizations about customers age, income, and gender:

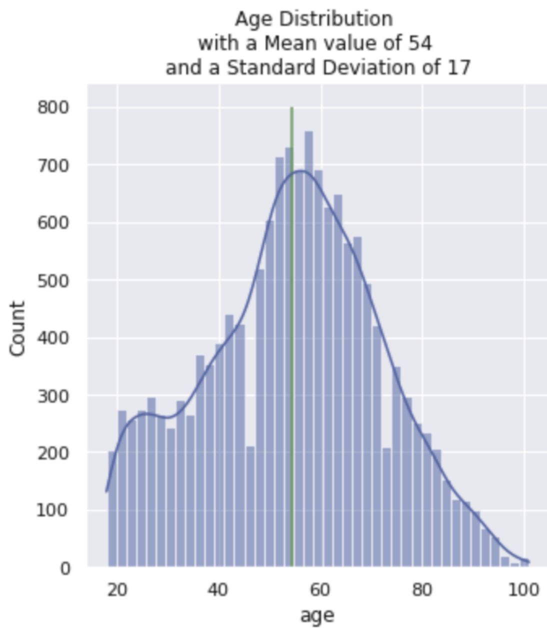

Figure 3: Age Distribution

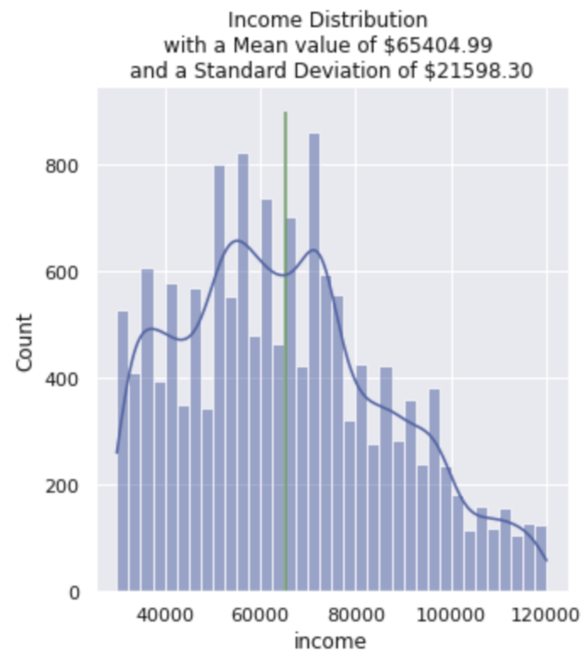

Figure 4: Income Distribution

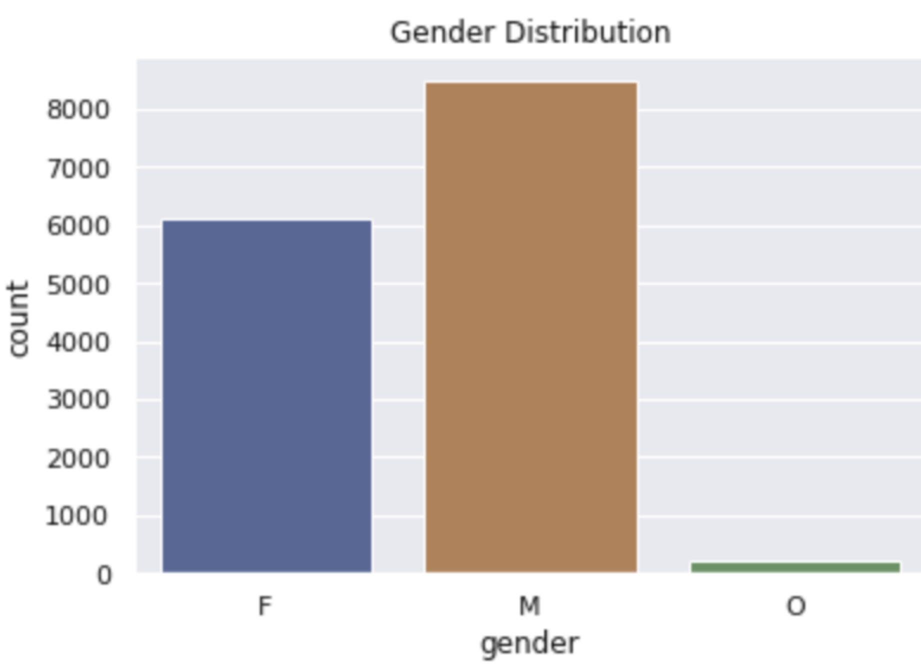

Figure 5: Gender Distribution

transcript

From initial exploration these are the observations:

- There are 306,534 rows and 4 columns.
- The dataset contains 4 columns:
    - person (string/hash): an id corresponds to a customer id.
    - event (string): one of four types; offer received, offer viewed, transaction, offer completed.
    - value (dictionary): different values depending on event type:
    - offer id (string/hash): an id corresponds to an offer id, not associated with any ”transaction” event.
    - amount (numeric): the money spent in ”transaction”.
    - reward (numeric): money gained/rewarded from ”offer completed”.
    - time (numeric): hours after start of test.
- The dataset contains no null values.
- All 3 offers were sent to customers almost equally.
- All 3 offers were not equally viewed by customers.
- BOGO and discount offers were completed by the customers with different interests.
- The dataset is balanced by looking at the value counts for all transcript completed offers.

	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
...	...	...	...	...
306529	b3a1272bc9904337b331bf348c3e8c17	transaction	{'amount': 1.5899999999999999}	714
306530	68213b08d99a4ae1b0dcb72aebd9aa35	transaction	{'amount': 9.53}	714
306531	a00058cf10334a308c68e7631c529907	transaction	{'amount': 3.61}	714
306532	76ddbd6576844afe811f1a3c0fbb5bec	transaction	{'amount': 3.5300000000000002}	714
306533	c02b10e8752c4d8e9b73f918558531f7	transaction	{'amount': 4.05}	714

306534 rows × 4 columns

Figure 6: transcript Dataset

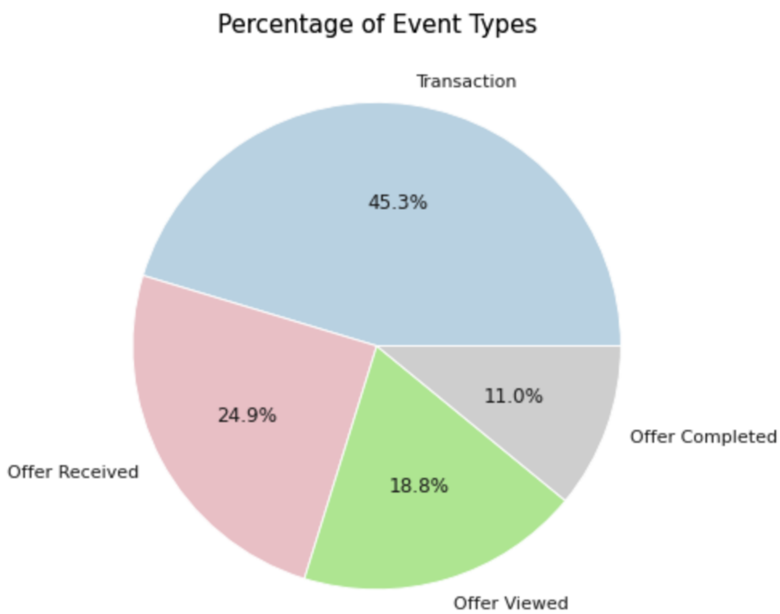

Figure 7: Percentage of Event Types

After exploring each dataset individually, we tried to explore the three datasets together. And by tracking the records of the most active customer for 30 days, we noticed:

- Not all received offers were viewed before completed, some customers completed the offer immediately after receiving it (check ids 113102 and 198780 in the below figure).
- Completed offers are always linked to a received offer and/or a viewed offer, so when combining profile, portfolio, and transaction datasets, received and viewed offer records for the same customer will hold the same information as completed offer of the same customer.
- So, to consider a customer as influenced by an offer, the customer must complete an offer. Thus I will only consider the completed offer in the final (merged) dataset.

113102	94de646f7b6041228ca7dec82adb97d2	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	336
152891	94de646f7b6041228ca7dec82adb97d2	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	408
164148	94de646f7b6041228ca7dec82adb97d2	offer viewed	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	408
168269	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 9.66}	414
171735	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 1.35}	420
171736	94de646f7b6041228ca7dec82adb97d2	offer completed	{'offer_id': 'fafdcd668e3743c1bb461111dcafc2a4...	420
177767	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 1.8399999999999999}	432
180401	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 0.93}	438
185283	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 2.79}	450
187480	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 1.7000000000000002}	456
193520	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 0.99}	474
198779	94de646f7b6041228ca7dec82adb97d2	transaction	{'amount': 1.4}	492
198780	94de646f7b6041228ca7dec82adb97d2	offer completed	{'offer_id': '0b1e1539f2cc45b7b9fa7c272da2e1d7...	492

Figure 8: Tracking Customer's Records

Methodology

This section shows all the processes taken to deal with the above mentioned datasets starting from cleaning them, feature engineering, and merging them into one dataset until splitting the data to training and testing datasets, training the models over training data, and finally evaluating the trained model over testing data.

1. **Data Cleaning**

portfolio

The dataset is almost clean, it has no null values or incomplete data. We only rename some column names and move the id column to the beginning of the dataset for better presentation.

profile

The profile dataset needs some cleaning process as it was noticed in the previous step that there are 2,175 null values in income, age, and gender columns. The following is done:

- Change the "118" value in age column to NaN.
- Fill the missing values in age by the mean.
- Fill the missing values in income by the mean.
- Fill the missing values in gender by "O".
- Rename some column names.
- Move the id column to the beginning of the dataset.

transcript

Transcript dataset has no null values, We only rename some column names and move the id column to the beginning of the dataset for better presentation.

2. Feature Engineering and Data Transformation

This section transforms the categorical features in all datasets and prepares the data for modeling step.

portfolio

For portfolio dataset, the following processes are considered:

- Replace the “offer_id” by more easy ids.
- Set “offer_id” as an index.
- Change the unit of “offer_duration” column from days to hours.
- Rename “offer_duration” column to “offer_duration_h” representing that the unit of measurement now is “hours”.
- Normalize “offer_difficulty”, “offer_reward”, and “offer_duration_h” features using the MinMaxScaler.
- Create dummy variables from the “channels” column using one-hot encoding then drop the “channels” column.
- Remove the email column.
- Replace the “offer_type” by integers representing each offer type as follows: 0 for bogo and 1 for discount.

The final dataset looks like:

offer_id	offer_reward	offer_difficulty	offer_duration_h	offer_type	mobile	social	web
8	1.0	0.50	0.571429	0	1	1	0
5	1.0	0.50	0.285714	0	1	1	1
7	0.5	0.25	0.571429	0	1	0	1
1	0.5	1.00	1.000000	1	0	0	1
2	0.3	0.35	0.571429	1	1	1	1
10	0.2	0.50	1.000000	1	1	1	1
9	0.5	0.25	0.285714	0	1	1	1
3	0.2	0.50	0.571429	1	1	0	1

Figure 9: portfolio Dataset after Feature Engineering Process

profile

For profile dataset, the following processes are considered:

- Replace the "customer_id" by more easy ids.
- Set "customer_id" as an index.
- Create dummy variables from the "gender" column using one-hot encoding then drop the "gender" column.
- convert "member_since" into a better format.
- normalize "age", "income", and "membership total days" features using the MinMaxScaler.

The final dataset looks like:

customer_id	age	income	F	M	O	membership_total_days	membership_year
6962	0.438476	0.393389	0	0	1	0.290181	2017
399	0.445783	0.911111	1	0	0	0.206253	2017
3747	0.438476	0.393389	0	0	1	0.007680	2018
7997	0.686747	0.777778	1	0	0	0.243006	2017
10736	0.438476	0.393389	0	0	1	0.195283	2017
...	...	...	...	...	...	...	...
7265	0.325301	0.266667	1	0	0	0.028524	2018
2888	0.518072	0.466667	0	1	0	0.007131	2018
106	0.373494	0.477778	0	1	0	0.299506	2017
10568	0.783133	0.222222	1	0	0	0.477784	2016
15178	0.530120	0.577778	1	0	0	0.202414	2017

17000 rows × 7 columns

Figure 10: profile Dataset after Feature Engineering Process

transcript

For transcript dataset, the following processes are considered:

- Drop transaction, offer received, and offer viewed records as we are concerned with completed offers only.
- Drop “event” column.
- Replace the “customer_id” with the easy ids created before.
- Pop the “value” column and create “offer_id” and “reward” columns, then drop the “value” column.
- Normalize “reward” and “time_h” features using the MinMaxScaler.
- Replace the'offer_id' with the ids created before.

The final dataset looks like:

	customer_id	time_h	offer_id	reward
12658	10702	0.0	3	0.000
12672	16909	0.0	10	0.000
12679	6535	0.0	7	0.375
12692	6863	0.0	8	1.000
12697	9577	0.0	5	1.000
...	...	...	...	...
306475	787	1.0	2	0.125
306497	11200	1.0	2	0.125
306506	12352	1.0	10	0.000
306509	8783	1.0	10	0.000
306527	2362	1.0	10	0.000

Figure 11: transcript Dataset after Feature Engineering Process

After cleaning and feature engineering all the three datasets, we combine them and do final analysis before the modeling step, the final merged dataset looks like the following:

	customer_id	time_h	offer_id	reward	offer_reward	offer_difficulty	offer_duration_h	mobile	social	web	age	income	F	M	O	membership_tot
0	10702	0.0	3	0.000	0.2	0.50	0.571429	1	0	1	0.289157	0.733333	0	1	0	0
1	16909	0.0	10	0.000	0.2	0.50	1.000000	1	1	1	0.253012	0.411111	1	0	0	0
2	6535	0.0	7	0.375	0.5	0.25	0.571429	1	0	1	0.409639	0.466667	0	1	0	0
3	6863	0.0	8	1.000	1.0	0.50	0.571429	1	1	0	0.228916	0.688889	0	1	0	0
4	9577	0.0	5	1.000	1.0	0.50	0.285714	1	1	1	0.361446	0.355556	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
33574	787	1.0	2	0.125	0.3	0.35	0.571429	1	1	1	0.457831	0.344444	0	1	0	0
33575	11200	1.0	2	0.125	0.3	0.35	0.571429	1	1	1	0.438476	0.393389	0	0	1	0
33576	12352	1.0	10	0.000	0.2	0.50	1.000000	1	1	1	0.438476	0.393389	0	0	1	0
33577	8783	1.0	10	0.000	0.2	0.50	1.000000	1	1	1	0.253012	0.100000	0	1	0	0
33578	2362	1.0	10	0.000	0.2	0.50	1.000000	1	1	1	0.361446	0.555556	1	0	0	0

33579 rows × 18 columns

Figure 12: Combined Dataset

The final dataset contains 33,579 rows and 18 columns, all rows representing completed offers where customers were influenced by the offer and benefited from it. We are concerned about the offer type as it is our target feature, and by analyzing the combined dataset, we noticed that it is balanced; 17,910 records are discount offers and 15,669 are BOGO offers.

3. Train-Test Data

Before splitting the dataset to training and testing datasets, we use a tree model to help identify the most relevant features for classification - see Figure 13. We then choose the top 10 features as they contain some offer information and customer demographic information. We also add the last gender type remaining "F". We checked the correlations between these features and excluded the ones with high correlation - see Figure 14. The final list of chosen features are: “offer_id”, “reward”, “offer_difficulty”, “offer_duration_h”, “mobile”, “social”, “web”, “F”, “M” and “O”.

Next, we then split the data to training dataset and testing dataset with 70/30 ratio, respectively.

```
Top features:
1. feature 3 (0.230780)
2. feature 4 (0.209380)
3. feature 6 (0.182128)
4. feature 2 (0.178748)
5. feature 5 (0.101929)
6. feature 7 (0.045987)
7. feature 9 (0.034668)
8. feature 8 (0.015977)
9. feature 12 (0.000139)
10. feature 14 (0.000083)
11. feature 1 (0.000064)
12. feature 11 (0.000046)
13. feature 15 (0.000033)
14. feature 10 (0.000015)
15. feature 16 (0.000012)
16. feature 13 (0.000009)
17. feature 0 (0.000001)
```

Figure 13: Top Features Importance

	offer_id	reward	offer_reward	offer_difficulty	offer_duration_h	mobile	social	web	F	M	O
offer_id	1.000000	0.106055	0.106055	-0.443461	-0.022170	0.499977	0.430963	-0.241416	-0.019209	0.006681	0.028963
reward	0.106055	1.000000	1.000000	0.070550	-0.466142	-0.011183	0.236036	-0.620092	0.059725	-0.031173	-0.065901
offer_reward	0.106055	1.000000	1.000000	0.070550	-0.466142	-0.011183	0.236036	-0.620092	0.059725	-0.031173	-0.065901
offer_difficulty	-0.443461	0.070550	0.070550	1.000000	0.617008	-0.864405	-0.314018	-0.061355	0.031767	-0.012487	-0.044557
offer_duration_h	-0.022170	-0.466142	-0.466142	0.617008	1.000000	-0.507796	-0.225641	0.064643	-0.019652	0.011358	0.019127
mobile	0.499977	-0.011183	-0.011183	-0.864405	-0.507796	1.000000	0.457760	-0.118285	-0.025023	0.011237	0.031845
social	0.430963	0.236036	0.236036	-0.314018	-0.225641	0.457760	1.000000	-0.258400	-0.014816	0.005854	0.020712
web	-0.241416	-0.620092	-0.620092	-0.061355	0.064643	-0.118285	-0.258400	1.000000	-0.030027	0.012854	0.039676
F	-0.019209	0.059725	0.059725	0.031767	-0.019652	-0.025023	-0.014816	-0.030027	1.000000	-0.907008	-0.209259
M	0.006681	-0.031173	-0.031173	-0.012487	0.011358	0.011237	0.005854	0.012854	-0.907008	1.000000	-0.221991
O	0.028963	-0.065901	-0.065901	-0.044557	0.019127	0.031845	0.020712	0.039676	-0.209259	-0.221991	1.000000

Figure 14: Correlation between Selected Features

4. Modeling

a. Benchmark Model

This is a classification problem where the model predicts the offer that influences a customer. Logistic Regression is used as a benchmark model. It is a supervised machine learning algorithm which is mostly used in classification problems.

Since the dataset we are dealing with is not large, we followed the recommendations provided by Rafal Alencar to control the regularization by means of the penalty and C parameters to deal with the overfitting [5]. We set the penalty to 'l1' and C to 0.1.

b. Other Proposed Models

Here we build several models such as K-Nearest Neighbors (KNN), decision tree, Support Vector Machine (SVM), XGBoost, and Light Gradient Boosting Machine (LightGBM), fit them on a training dataset and predict testing data.

We also followed the recommendations suggested by Rafal in adjusting some of the models parameters to help in dealing with overfitting problem. For tree-based models like XGBoost, decision tree, and LightGBM we can control the overfitting by tuning a series of parameters such as the maximum depth by making the tree short [5][6].

5. Evaluation

We evaluate the benchmark model as well as the proposed models based on accuracy, precision, recall, F1, and Confusion Matrix. A summary of results for logistic regression, K-Nearest Neighbors (KNN), decision tree, Support Vector Machine (SVM), XGBoost, and LightGBM is provided in Figure 12. Confusion Matrix plots for all models are shown below.

	Logistics Regression (Benchmark Model)	K-Nearest Neighbors	DT	SVM	XGB	LightGBM
Accuracy	1.0	1.0	0.903514	1.0	1.0	1.0
Missclassification	0.0	0.0	0.096486	0.0	0.0	0.0
TP	4430.0	4430.0	3620.000000	4430.0	4430.0	5298.0
TN	3965.0	3965.0	3965.000000	3965.0	3965.0	4776.0
FP	0.0	0.0	0.000000	0.0	0.0	0.0
FN	0.0	0.0	810.000000	0.0	0.0	0.0
Precision 0	1.0	1.0	0.830366	1.0	1.0	1.0
Precision 1	1.0	1.0	1.000000	1.0	1.0	1.0
Recall 0	1.0	1.0	1.000000	1.0	1.0	1.0
Recall 1	1.0	1.0	0.817156	1.0	1.0	1.0
F1-score 0	1.0	1.0	0.907323	1.0	1.0	1.0
F1-score 1	1.0	1.0	0.899379	1.0	1.0	1.0

Figure 15: Evaluation Results

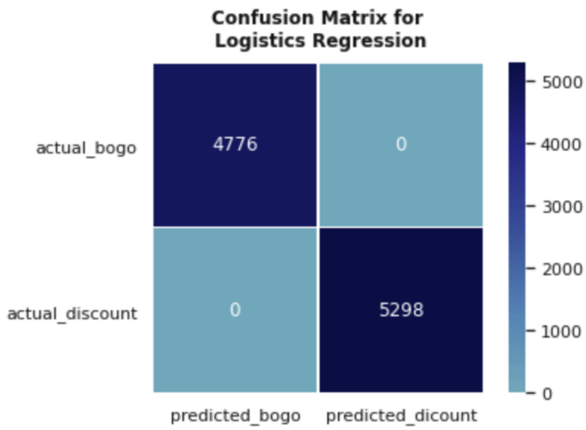

Figure 16: Logistics Regression Confusion Matrix

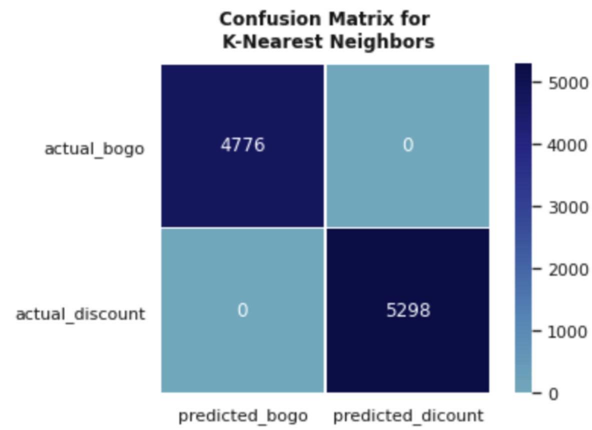

Figure 17: K-Nearest Neighbors Confusion Matrix

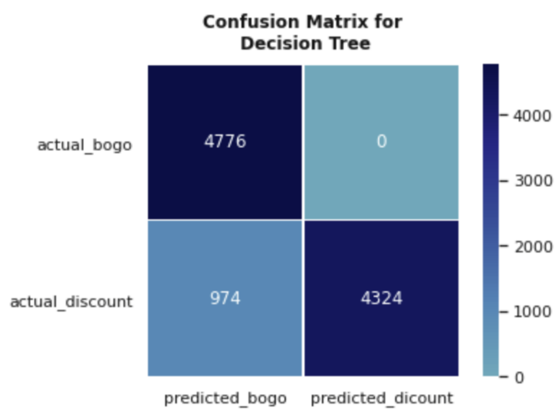

Figure 18: Decision Tree Confusion Matrix

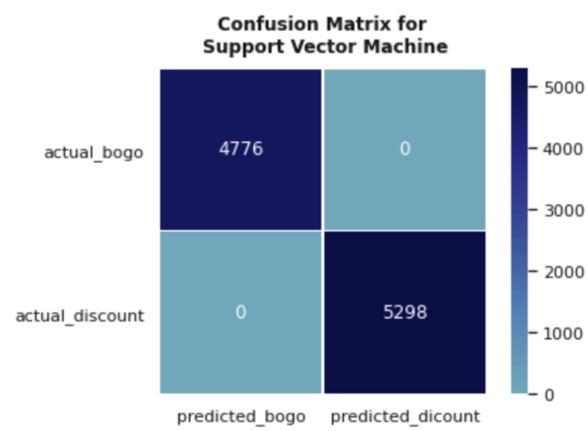

Figure 19: Support Vector Machine Confusion Matrix

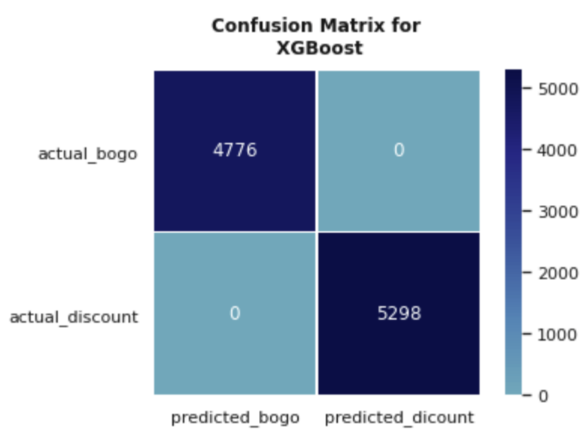

Figure 20: XGBoost Confusion Matrix

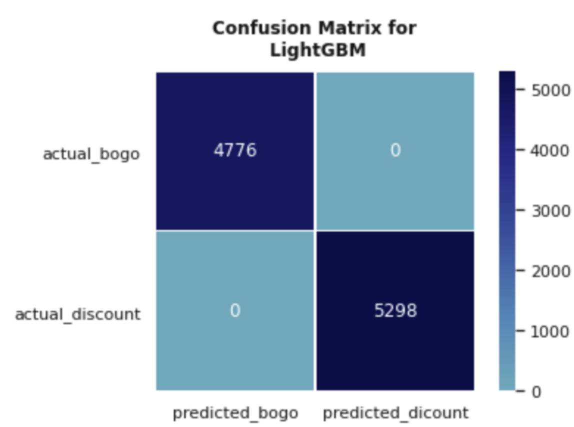

Figure 21: LightGBM Confusion Matrix

6. **Results**

Section 5 provides the performance results of the benchmark model (logistic regression) and the other models (K-Nearest Neighbors (KNN), decision tree, Support Vector Machine (SVM), XGBoost, and LightGBM). As shown, most models reported an accuracy of 100% with correct classification of both offer types: 0 for bogo (buy-one-get-one) and 1 for discount. Decision tree on the other hand reported an accuracy of 90% misclassifying 810 discount offer.

We think these high results were achieved due to the small dataset we are dealing with; we are only dealing with 33,579 rows and 18 columns (23,505 for training and 10,074 for testing).

To avoid overfitting, I will choose the model with the lowest accuracy score over the testing dataset; the decision tree, the model reported an accuracy of 90% which is still considered a good result.

Model Refinement

Although the results obtained from applying different classification algorithms were high, it might be worth trying other algorithms that work best with small datasets. In general, small datasets require simple and low complex/or high bias machine learning algorithms to avoid overfitting [6]. An example of such an algorithm is Naive Bayes, it is among the simplest classifiers which proved learning well from small dataset.

Conclusion

For this project, we dealt with three different datasets for Starbucks loyalty program application. First, we explored and analyzed them, then we cleaned them by dealing with missing values and renaming some features, next, we did some feature engineering and data transformation work for all the three datasets. Once they became ready for the modeling step, we combined them based on offer id and customer id to get the behaviors' records of all customers with offer and customer information. Later, we moved to the modeling step by building various models, then using several measurements, we evaluated the built models over the testing data and obtained a high accuracy score.

Future work

The portfolio dataset contains three offer types, each of which contain several forms. That is, buy-one-get-one offer has 4 forms, discount offer has 4 forms, and informational offer has 2 forms. Each form differs from another by the offer duration, reward obtained, difficulty, and the channel used for sending the offer. So, I am curious if there is a different behavior for each form of offer, that is, dealing with multi-class classification problem.

We tried building this multi-class classification scenario and got some initial results; however, more investigation and good decisions need to be made about the suitable model, the best parameters, and the good metrics for such a problem.

Below is some of the work done, the first figure shows the target “offer_type” with 8 different classes; 4 for bogo offer and 4 for discount offers. Informational offers were removed as no record was found for them. It was noticed also that the dataset is balanced among these types.

- This is the combined dataset with muli-class labels.

reward	offer_reward	offer_difficulty	offer_duration_h	mobile	social	web	age	income	F	M	O	membership_total_days	membership_year	offer_type
0.000	0.2	0.50	0.571429	1	0	1	0.289157	0.733333	0	1	0	0.505211	2016	3
0.000	0.2	0.50	1.000000	1	1	1	0.253012	0.411111	1	0	0	0.121229	2017	10
0.375	0.5	0.25	0.571429	1	0	1	0.409639	0.466667	0	1	0	0.027976	2018	7
1.000	1.0	0.50	0.571429	1	1	0	0.228916	0.688889	0	1	0	0.239715	2017	8
1.000	1.0	0.50	0.285714	1	1	1	0.361446	0.355556	0	1	0	0.579813	2015	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.125	0.3	0.35	0.571429	1	1	1	0.457831	0.344444	0	1	0	0.150850	2017	2
0.125	0.3	0.35	0.571429	1	1	1	0.438476	0.393389	0	0	1	0.304992	2017	2
0.000	0.2	0.50	1.000000	1	1	1	0.438476	0.393389	0	0	1	0.300055	2017	10
0.000	0.2	0.50	1.000000	1	1	1	0.253012	0.100000	0	1	0	0.015908	2018	10
0.000	0.2	0.50	1.000000	1	1	1	0.361446	0.555556	1	0	0	0.115195	2017	10

Figure 22: Combined Dataset with Multi-Class Classification

- The below figure is the result obtained with logistic regression, knn, and decision tree models. However, as said before, a further investigation need to be made to determine the best options for multi-class classification problem.

	Logistics Regression	knn	DT
Accuracy	1.0	1.0	0.564503
TP	1304.0	1304.0	1304.000000
TN	810.0	810.0	0.000000
FP	0.0	0.0	810.000000
FN	0.0	0.0	0.000000

Figure 23: Muli-Class Classification Evaluation Results

References

[1] Nikolina Jeric. 30+ starbucks statistics and facts with pumpkin spice, Apr 2021.

[2] The Manifest. The success of starbucks app: A case study, June 2018.

[3] Riley Panko. How customers use food delivery and restaurant loyalty apps, May 2018.

[4] Mishra, A., 2021. *Metrics to Evaluate your Machine Learning Algorithm*. [online] Medium. Available at: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234> [Accessed 23 December 2021].

[5] ALENCAR, R., 2021. *Dealing with very small datasets*. [online] Kaggle.com. Available at: <https://www.kaggle.com/rafjaa/dealing-with-very-small-datasets> [Accessed 23 December 2021].

[6] ALENCAR, R., 2021. *Dealing with very small datasets*. [online] Kaggle.com. Available at: <https://www.kaggle.com/rafjaa/dealing-with-very-small-datasets> [Accessed 23 December 2021].