Capstone Proposal: Starbuck's Challenge

John Hodge
jah70.udacity@gmail.com

April 19, 2021

1 DOMAIN BACKGROUND

Starbucks is the world's largest coffeehouse chain and Fortune 500 company in the United States. The company's headquarters are in Seattle, Washington, where it was founded in 1971. As one of the world's largest companies, Starbucks operates over 30,000 stores and serves millions of customers worldwide.

To retain customer loyalty and increase business success, Starbucks operates a customer rewards program through the Starbucks mobile app. The company's mobile app allows registered customers to place pick-up orders, pay inside stores, and earn reward points. In-app marketing through the mobile app is a critical component of Starbucks' direct marketing strategy. Starbucks sends customers promotional offers through the mobile app once every couple of days. These promotions include drink advertisements, discount offers, and buy one get one free (BOGO) offers.

To maximize the effectiveness of these promotional offers, not every customer receives the same promotional offer. Instead, Starbucks tailors promotions and advertisements to the unique characteristics of individual customers and their customer segment. In recent years, machine learning techniques have produced state-of-the-art systems for recommendation [1, 2], customer segmentation [3], consumer demand forecasting [4, 5], and forecasting consumer behavior based on promotional marketing [6, 7]. This project focuses on using machine learning and data science methods to predict customer responses to tailored marketing and promotional offers.

As a coffee enthusiast, Starbucks rewards member, and user of the Starbucks mobile app, this capstone project stands out to me as exciting data science and machine learning investigation. Additionally, learning how a personalized marketing campaigns works allows me to better understand how quantitative methods are used to increase customer satisfaction and business success in future engineering projects.

2 Problem Statement

As stated in the Starbucks' Capstone Challenge overview, the task is to use the data to identify which groups of people are most responsive to each offer and how best to present each type of offer. Data analytics discovers the hidden traits that influence their purchasing decisions and responses to promotional offers for each customer segment in the simulated dataset. In this project, a machine learning model will be developed to predict how much a customer will spend based on offer type, demographics, and their responses to previous offers.

The goal is to determine what type of advertisement or promotional offer will achieve the highest return on investment (ROI) for a given customer over a set period. The ROI of an ad is the amount a customer spends minus the cost of the promotional discount. The business needs to understand the best type of marketing to serve each customer segment and accurately predict the ROI of each advertisement. Implicitly, this also predicts the people's responsiveness to each offer type as it will have either a positive, negative, or neutral expected value on how much they spend.

3 Datasets and Inputs

The structure of the dataset provided Starbucks Capstone project notebook is structured as follows. The data is contained in three files:

- *portfolio.json* - containing offer ids and meta data about each offer (duration, type, etc.) (See Fig. 3.1)
- *profile.json* - demographic data for each customer (See Fig. 3.2)
- *transcript.json* - records for transactions, offers received, offers viewed, and offers completed (See Fig. 3.3)

The three types of offers presented in the *_type* column of *portfolio.json* are:

- 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 offer: there is no reward, but neither is there a requisite amount that the user is expected to spend.

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

portfolio.json
Size: 10 offers by 6 fields

```
In [2]: portfolio.head(10)
```

Out[2]:

	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5
5	[web, email, mobile, social]	7	7	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	3
6	[web, email, mobile, social]	10	10	fafdcd668e3743c1bb461111dcafc2a4	discount	2
7	[email, mobile, social]	0	3	5a8bc65990b245e5a138643cd4eb9837	informational	0
8	[web, email, mobile, social]	5	5	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5
9	[web, email, mobile]	10	7	2906b810c7d4411798c6938adc9daaa5	discount	2

```
In [3]: print("Portfolio Data Dimensions: ", portfolio.shape)

Portfolio Data Dimensions:  (10, 6)
```

Figure 3.1: The first 10 rows and the dataset dimensions of *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
Size: 17,000 users by 5 fields

- *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
Size: 306,534 offers by 4 fields

- *event* (str) - record description (ie transaction, offer received, offer viewed, etc.)

```
In [4]: profile.head(10)
```

Out[4]:

	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN
5	68	20180426	M	e2127556f4f64592b11af22de27a7932	70000.0
6	118	20170925	None	8ec6ce2a7e7949b1bf142def7d0e0586	NaN
7	118	20171002	None	68617ca6246f4fbc85e91a2a49552598	NaN
8	65	20180209	M	389bc3fa690240e798340f5a15918d5c	53000.0
9	118	20161122	None	8974fc5686fe429db53ddde067b88302	NaN

```
In [5]: print("Profile Data Dimensions: ", profile.shape)

Profile Data Dimensions:  (17000, 5)
```

Figure 3.2: The first 10 rows and the dataset dimensions of *profile.json*.

- *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

For the forecasting (predict spending) and classification (predicting best offer) problems, I will join the three datasets into a unified dataset that combines the customer profile and offer characteristics with the transcript event and transaction data. Principle component analysis (PCA) creates customer segments based on their customer profiles. For the forecasting problem, the inputs are customer segment id and offer characteristics and the output will be the the transaction amount within the offer duration. The input data will be the customer segment id, and the output data will be the offer id for the classification problem.

Training, validation, and test datasets are created by randomly splitting up "offer received" events in the *transcript.json* dataset. The dataset is augmented by including a column of transaction amounts if a transaction occurs within the offer period. If not transaction occurs, the transaction amount is 0. The dataset will also have a cost column with the reward data from *portfolio.json*. I add a marketing-adjusted revenue column to the *transcript.json* dataset that subtracts the cost of the reward from the non-zero (positive) transaction amounts.

If I choose a recurrent neural network (RNN) as the model, I will format the data as a time series for each customer. In this scenario, the data is transformed by grouping transactions *person* (customer id) and sorting them by time. This process allows a time series of events (either an offer id or transaction amount) to be input to the RNN.

```
In [6]: transcript.head(10)
```

Out[6]:

	event	person	time	value
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
2	offer received	e2127556f4f64592b11af22de27a7932	0	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}
5	offer received	389bc3fa690240e798340f5a15918d5c	0	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}
6	offer received	c4863c7985cf408faee930f111475da3	0	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}
7	offer received	2eeac8d8feae4a8cad5a6af0499a211d	0	{'offer id': '3f207df678b143eea3cee63160fa8bed'}
8	offer received	aa4862eba776480b8bb9c68455b8c2e1	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
9	offer received	31dda685af34476cad5bc968bdb01c53	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}

```
In [7]: print("Transcript Data Dimensions: ", transcript.shape)

Transcript Data Dimensions:  (306534, 4)
```

Figure 3.3: The first 10 rows and the dataset dimensions of *transcript.json*.

4 Solution Statement

To discover what type of advertisement or promotional offer will generate the highest ROI for each customer, a machine learning model will be trained to predict how much a customer will spend, based on offer type, demographics, and their responses to previous offers. In recent years, deep neural networks (DNNs) have become the state-of-the-art for similar customer forecasting [8] and recommendation systems [1, 2]. The machine learning model will likely be a deep neural network (DNN) and potentially a RNN. Secondly, I will also predict which type of offer is best for each customer if time permits. This is a classification problem that will likely use a DNN as the predictive model.

5 Benchmark Model

The XGBoost algorithm will be used as the benchmark model to compare our model's prediction performance. XGBoost is an open-source and efficient implementation of the gradient boosted tree algorithm. Gradient boosting is a supervised learning algorithm that was used in one of the course lessons to predict housing price data. If time permits, the linear and logistical regression algorithms will also be used as a benchmark model. Amazon Web Services (AWS) Sagemaker provides simple implementations of the XGBoost and Linear Learner algorithms. These boosted tree and regression algorithms can also be implemented in Python using scikit-learn.

6 Evaluation Metrics

This project builds a predictive model of how much a customer will spend in response to an advertisement or promotional offer. Since this is a regression problem, the mean squared error (MSE) between the amount that the model predicted a customer would spend based on the offer type and how much they spend will be the primary metric of model evaluation in this study. The explained variance score and R2 score are additional evaluation metrics under consideration.

Additionally, the precision, accuracy, and recall of the the model will be used as evaluation metrics to determine which type of offer (discount, BOGO, or informational) is best for each customer. The F1 score, a weighted average of precision and recall, will be used as the primary evaluation metric to determine the best model. Additionally, the area under the receiving operating characteristic (ROC) curve (AUC) will also be considered as an evaluation metric.

7 Project Design

The project will be performed and documented in a Jupyter notebook environment for transparency and repeatability. The project will follow a standard machine learning workflow:

I. Data Preparation: Clean-up data if necessary for data modeling, visualization, and training purposes.

II. Data Exploration: Perform an exploratory analysis of the dataset, including data visualization, to better understand the contents and distributions of data in the dataset. This investigation of the dataset will provide additional insight into the most appropriate type of predictive model for this study.

III. Data Transformation: Combine different sources of data if necessary and create the target variable for training.

IV. Develop & Train Model: Build a predictive model by experimenting with different model architectures and performing hyperparameter tuning on the most promising model architecture to optimize the model and achieve the best training results.

V. Model Validation & Evaluation: The model predictions will be evaluated and compared to the benchmark model.

VI. Documentation: The project results will be summarized and described in a detailed blog post.

This capstone project will not include model deployment as it is beyond the scope of the capstone project requirements.

References

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[2] R. Mu, “A survey of recommender systems based on deep learning,” *IEEE Access*, vol. 6, pp. 69 009–69 022, 2018.

[3] K. K. Tsiptsis and A. Chorianopoulos, *Data mining techniques in CRM: Inside customer segmentation*. John Wiley & Sons, 2011.

[4] R. Law and N. Au, “A neural network model to forecast japanese demand for travel to hong kong,” *Tourism Management*, vol. 20, no. 1, pp. 89–97, 1999.

[5] G. P. Zhang, *Neural networks in business forecasting*. IGI Global, 2004.

[6] A. Y. L. Chong, E. Ch’ng, M. J. Liu, and B. Li, “Predicting consumer product demands via big data: The roles of online promotional marketing and online reviews,” *International Journal of Production Research*, vol. 55, no. 17, pp. 5142–5156, 2017.

[7] Z. Zhao, J. Wang, H. Sun, Y. Liu, Z. Fan, and F. Xuan, “What factors influence online product sales? online reviews, review system curation, online promotional marketing and seller guarantees analysis,” *IEEE Access*, vol. 8, pp. 3920–3931, 2019.

[8] A. L. Loureiro, V. L. Miguéis, and L. F. da Silva, “Exploring the use of deep neural networks for sales forecasting in fashion retail,” *Decision Support Systems*, vol. 114, pp. 81–93, 2018.