A Case Study on Starbucks Data

Capstone Project for Udacity ML Engineer Nanodegree

Muhammet Çelik - 06.07.2021

1. Introduction

Starbucks Corporation is an American multinational chain of coffeehouses and roastery reserves headquartered in Seattle, Washington. As the world's largest coffeehouse chain, Starbucks is seen to be the main representation of the United States' second wave of coffee culture. As of September 2020, the company had 32,660 stores in 83 countries, including 16,637 company operated stores and 16,023 licensed stores. Of these 32,660 stores, 18,354 were in the United States, Canada, and Latin America [1].

The rapid spread of mobile applications have changed not only lives of people, but also the way of doing business. Advertisement business has altered its shape into online advertisement. On contrary to conventional methods, online advertisements offer more targeted customers to the companies. This phenomenon is also valid in the scope of this project. The aim of this project, by using the data gathered by Starbucks App, to develop a smart model making promotion offers to customers according to their demographic information and coffee drinking habits.

1.1. Problem Statement

Starbucks demands an intelligent model deciding whether to show the online offer or not to each individual customer. Moreover, the company also wants to show the best offer to them, increasing the offer acceptance probability.

To deal with this problem, we are going to use the customers historical data and build a model. Performance evaluation of the model can be made by observing the past offers and their acceptance and rejection states. The solution will be based on the demographic features, the behavioral features and also the promotion features.

1.2. Target Definition

In the ML perspective, a model will be developed deciding the end result of the offer when a customer receives an offer.

There are three cases:

- Customer receives, views and completes the offer.
    - **Class 0:** Offer Received -> Offer Viewed -> Offer Completed
- Customer receives and views the offer, but does not complete it.
    - **Class 1:** Offer Received -> Offer Viewed
- Customer receives the offer but does not neither view or complete it.
    - **Class 2:** Offer Received

Assumptions

The main goal will be a classification problem. By using the given data sets, the feature set will be engineered in consistent with the cases above. In other words:

- To complete an offer, a customer should receive and view the offer, respectively.
- Offer states should be aligned on the time. A customer cannot view offer before receiving it.
- Once an offer is viewed or completed, then the same offer cannot be viewed and completed again.
- A customer can receive the same offer in different times, this cancels the offer had sent in the past.
- If the customer receives and completes an offer without viewing, view time is assumed as the same as completion time.

1.3. Data Sets & Inputs

This dataset contains simulated data that mimics customer behavior 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 merely an advertisement for a drink or an actual offer such as a discount or BOGO (Buy-One-Get-One free). Some users might not receive any offer during certain weeks. Not all users receive the same offer, and that is the challenge to solve with this dataset [2].

There are three available data sources: **portfolio.json, profile.json, transcript.json**. The detailed information has been given in Table 1.

1.4. Benchmark Models & Evaluation Metrics

As a benchmark model, a dummy model, a KNN model and a Logistic Regression with default parameters are built. Due to the fact that the input data set will be tabular, Decision Tree and Random Forest models is expected to perform better. Hyperparameter tuning for these models is made to find the best. For performance, both false negatives and false positives are important, also the target is multiclass, weighted F1 score is the base metric for performance evaluation.

Data Source	Source Explanation	Variable Name	Variable Type	Variable Explanation
portfolio .json	**10** records, contains offer ids and metadata about each offer (duration, type, etc.)	id	string	offer id
		offer_type	string	type of offer
		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	channels for showing offers
profile .json	**17000** records, contains demographic data for each customer	age	int	age of the customer
		became_member_on	int	date when customer created an app account
		gender	string	gender of the customer
		id	string	customer id
		income	float	customer's income
transcript .json	**306533** records, contains records for transactions, offers received, offers viewed, and offers completed	event	string	record description (ie transaction, offer received, offer viewed)
		person	string	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

Table 1: Data sources and their explanations

2. Data Preparation

Since the data is introduced briefly in Section 1.3., transformation of the data setes will be presented. Explorations and insights will be the key of data preparation. More details can be found on notebooks.

2.1. Portfolio

```
portfolio.head()
```

	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

Processes

- **id** is renamed to **offer_id.**
- Channel list is broaddcasted as binary variables: **web**, **mobile**, **social**, **email**.
- **email** column is dropped due to that it is constant.

```
portfolio_ = portfolio_.drop(["channels", "email"], 1)
portfolio_.head()
```

	offer_id	offer_type	difficulty	duration	reward	web	mobile	social
0	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10	7	10	0	1	1
1	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10	5	10	1	1	1
2	3f207df678b143eea3cee63160fa8bed	informational	0	4	0	1	1	0
3	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5	7	5	1	1	0
4	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	20	10	5	1	0	0

2.2. Profile

```
profile.head()
```

	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

Processes

- **id** is renamed to **customer_id.**
- Customers are grouped with **became_member_on** column. Limits are defined by graphical analysis shown in Figure 1. -> **membership**
- **age** and **income** is bucketed into 12 and 6 groups respectively.
- Dataset has 2135 NULL values over 17000 records. These rows are imputed with a Bayesian approach.
- More details can be found in **starbucks_capstone_data_exploration** notebook.

```
profile_ = pd.concat([df_normal, df_null], ignore_index=True)
profile_.head()
```

	customer_id	membership	gender	age_bucket	income_bucket	was_null_profile
0	0610b486422d4921ae7d2bf64640c50b	mid	F	53_59	100k_120k	0
1	78afa995795e4d85b5d9ceeca43f5fef	mid	F	72_77	100k_120k	0
2	e2127556f4f64592b11af22de27a7932	new	M	66_71	60k_75k	0
3	389bc3fa690240e798340f5a15918d5c	new	M	60_65	50k_60k	0
4	2eeac8d8feae4a8cad5a6af0499a211d	new	M	53_59	50k_60k	0

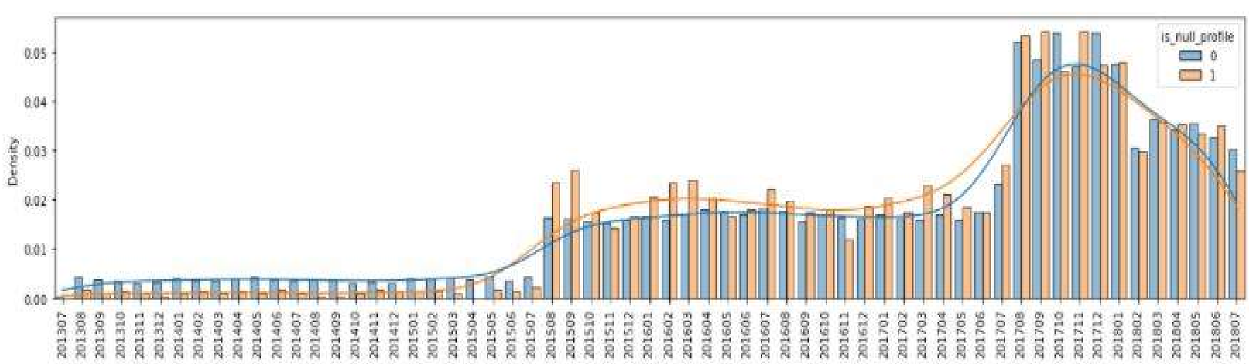

Figure 1: become_member_on monthly aggregated count plo**t**

2.3. Transcript

This data set is split into two groups: Transactions and Offers. These two data set is preprocessed separately.

```
transcript.head()
```

	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

2.3.1. Transaction

This data set includes customer_id, time and transaction amount

__Processes__

- **person** is renamed to **customer_id**.
- By using time and amount column, different features are created:
  - **total_trans_amount**: Total payments that the customer made.
  - **total_trans_count**: Number of transactions that the customer made.
  - **avg_trans_amount**: Average amount of the customer's transactions.

	customer_id	time	amount	total_trans_amount	total_trans_count	avg_trans_amount
0	0009655768c64bdeb2e877511632db8f	228	22.16	22.16	1	22.160
1	0009655768c64bdeb2e877511632db8f	414	8.57	30.73	2	15.365
2	0009655768c64bdeb2e877511632db8f	528	14.11	44.84	3	14.947
3	0009655768c64bdeb2e877511632db8f	552	13.56	58.40	4	14.600
4	0009655768c64bdeb2e877511632db8f	576	10.27	68.67	5	13.734

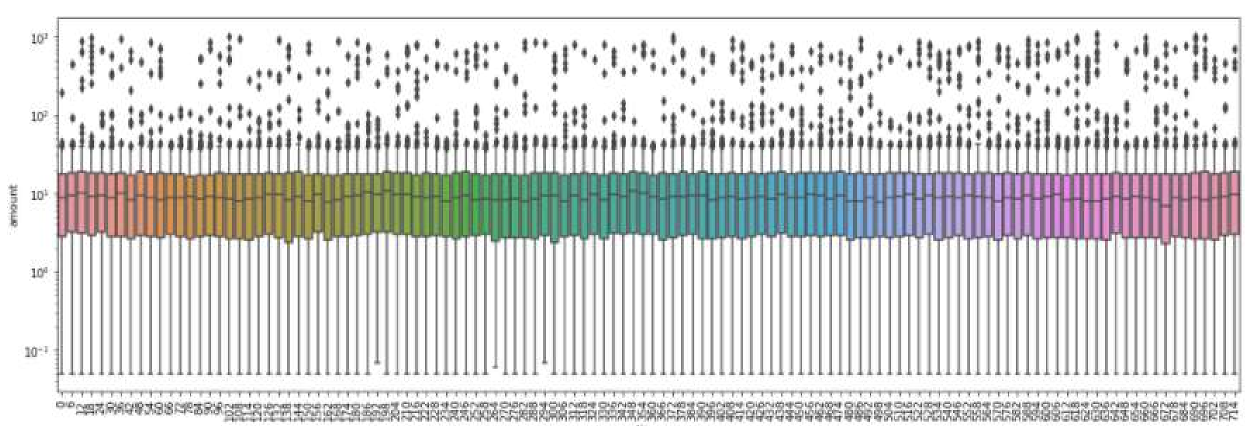

Figure 2: Transaction amount over time (log scale)

2.3.2. Offer

This data set is the basis of the model data set. It includes offer related information. For our case, when an offer is received, that creates a sample for the model. For each customer and for each offer, there is a result:

- Offer viewed and completed (class 0)
- Offer viewed but not completed (class 1)
- Offer neither viewed nor completed. (class 2)

Due to this setting, the data is cleaned and imputed in that way. Offer chains are created, then these chains are decomposed. Each received offer is matched with viewed and completed properly. recurrence column shows the number of the same offer received by the same customer. More information can be found in preparation notebook.

	customer_id	offer_id	recurrence	received	target
0	0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5	0	576	0
1	0009655768c64bdeb2e877511632db8f	3f207df678b143eea3cee63160fa8bed	0	336	1
2	0009655768c64bdeb2e877511632db8f	5a8bc65990b245e5a138643cd4eb9837	0	168	1
3	0009655768c64bdeb2e877511632db8f	f19421c1d4aa40978ebb69ca19b0e20d	0	408	0
4	0009655768c64bdeb2e877511632db8f	fafdcd668e3743c1bb461111dcafc2a4	0	504	0

	customer_id	offer_id	time_event
0	0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5	576A-576C
1	0009655768c64bdeb2e877511632db8f	3f207df678b143eea3cee63160fa8bed	336A-372B
2	0009655768c64bdeb2e877511632db8f	5a8bc65990b245e5a138643cd4eb9837	168A-192B
3	0009655768c64bdeb2e877511632db8f	f19421c1d4aa40978ebb69ca19b0e20d	408A-414C-456B
4	0009655768c64bdeb2e877511632db8f	fafdcd668e3743c1bb461111dcafc2a4	504A-528C-540B

Figure 3: Offer Chains

3. EDA & Feature Engineering

At this stage, all features are joined to the base table. Correlation analysis is made and correlated features are dropped. Categorical features are converted into binary variables by using One-Hot Encoding. Feature effects on target is investigated. Some findings can be sorted as:

- Offer completion probability of mid-level members are slightly higher.
- As age increases, offer completion probability increases.
- As income increases, similar to age, offer completion probability increases.
- Being a null profile severely decreases offer completion probability.
- Hence the time difference increases, offer view probability increases.
- mobile and social channels have a significant impact on offer completion.
- recurrence has no significant effect on the distribution of the target.

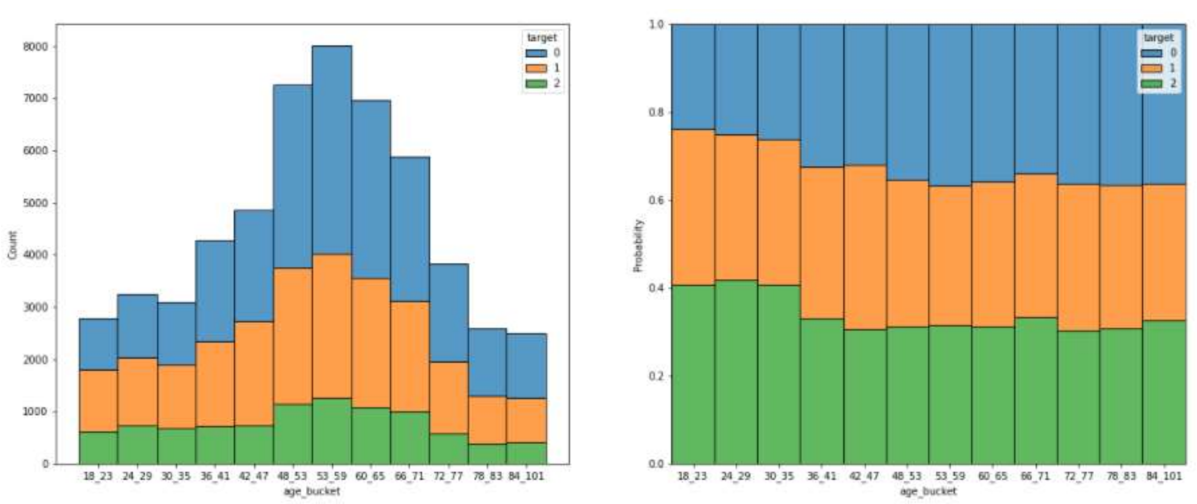

Figure 4: As age increases, offer completed probability inreases slightly.

The final dataset consists of 55325 records with 47 feature columns.

4. Modelling & Evaluation

A dummy classifier, logistic regression models and knn is used as benchmark models. Decision Tree and Random Forest model is expected to perform better, for this reason a grid search is performed targeting to increase weighted F1 score with cross-validation with 5 folds. Performance is obtained from both train and test datasets.

- Regarding performance, the fundamental metric is f1_score weighted.
- Weighted f1_score provides the general performance of the model considering sample size of the classes, also macro average is calculated for insight.

4.1.1. Train Performance

- Decision Tree & Random Forest without CV performed well on train. It seems that they overfitted.
- Dummy Classifier is the worst.

4.1.2. Test Performance

- The best model is tuned Random Forest Model.
- Tuned Decision Tree and Random Forest with default parameters are performed as well.

4.1.3. Feature Importance

- For Tuned Random Forest Model
    - Transaction related features are very important for decisions.
    - Reward amount is also important.
    - Age and income related features are not very decisive.

- For Tuned Decision Tree Model
    - The model cares to similar features as Tuned Random Forest.
    - Social channel is slightly important.

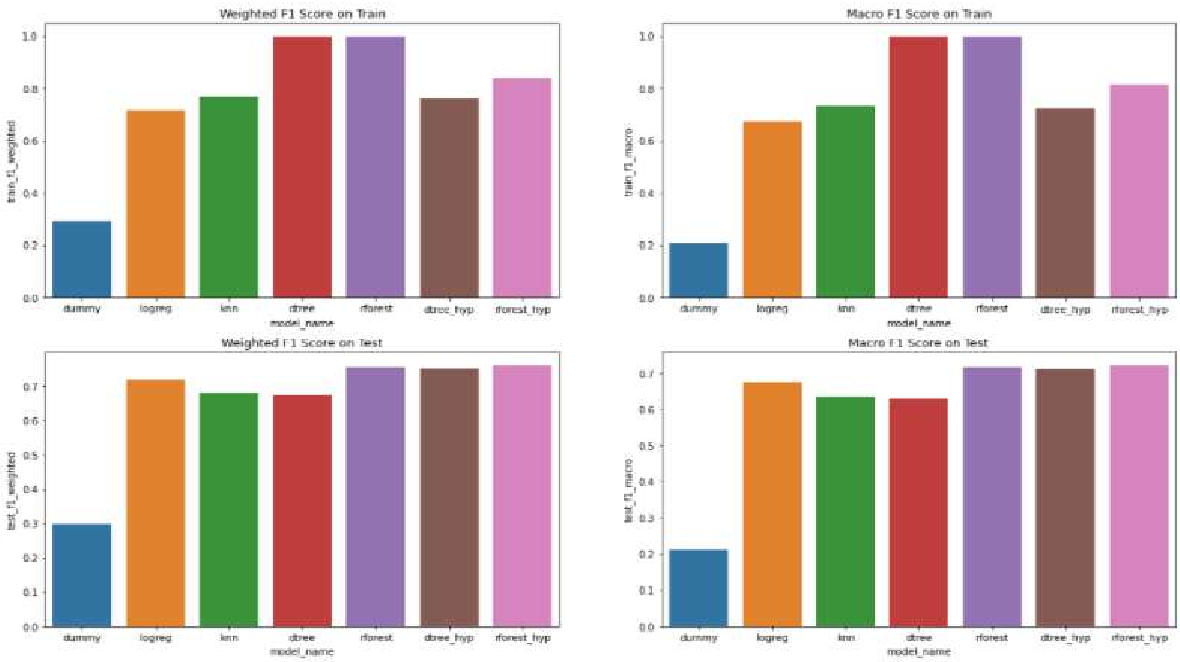

Figure 5: Model Performance Comparison

	model_name	train_f1_weighted	train_f1_macro	test_f1_weighted	test_f1_macro
0	dummy	0.290593	0.210260	0.299326	0.212802
1	logreg	0.716643	0.672570	0.720675	0.673577
2	knn	0.768696	0.734681	0.682594	0.634218
3	dtree	1.000000	1.000000	0.674317	0.629309
4	rforest	0.999977	0.999982	0.755723	0.715699
5	dtree_hyp	0.761987	0.724036	0.752178	0.710659
6	rforest_hyp	0.839505	0.813823	0.761401	0.721633

5. References

[1] https://en.wikipedia.org/wiki/Starbucks

[2] Udacity ML Engineer Nanodegree – Starbucks Project Workspace