Capstone Project

Machine Learning Engineer
Nanodegree

Dipan Sutradhar
May 20, 2020

Starbuck's Capstone Challenge

Definition

Project Overview

Starbucks Capstone Challenge is a capstone project of the Udacity Machine Learning Nanodegree program. This project addresses a classification challenge that predicts the customers' demographics influence on customers' responds to Starbucks mobile rewards program. The data set includes Starbucks mobile app reward data with simulated customer demographics.

- Portfolio: Currently available offer details
- Profile: Customer demographic details
- Transcript: Event details on customers' action

Problem Statement

Like any other campaigning program, the success of the Starbucks reward program depends on customers' participation in the program. It is critical for the reward program that offered reward is accepted by customers. So Starbucks wants to analyze and identify if there is any influence on customers' demographics. Eventually, develop a model to predict the customers' responsiveness based on their demographic and offer details. This will help Starbucks to back their advertisement decision by data.

Metrics

As any binary classification model, these models also predict whether customers will accept the offer or not. Mathematical representation of this result is either 1 or 0 for a given set of features. I use a set of test data so that I can use the result set to represent the performance of the models on percentage.

Analysis

Data Exploration

Starbucks datasets separated into three such as portfolio, profile, and transcript in JSON format. The schema and sample data is shown below:

- **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)

	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

Figure 1: Portfolio data

Above data is about Starbucks 10 available offer details. Data represents the features of these offers such as what are the channels each offer can reach to the customer, how difficult the offer to avail for customer, how long customer can wait before opt in, what type of offer is this, and lastly what is the monetary value of the reward. For instance, a customer may prefer discount over bogo (i.e. buy one get one free). But this data has list data for channels and categorical data for offer type which needs to be clean.

- **profile.json**
  - 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

	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

Figure 2: Profile data

Above subset of profile data represents customer demographics. Models are trained on the demographic of customers such as how to old the customer is, how much a customer earns, or even how long a customer is signed up into reward program. This dataset has missing value such as NaN in income column and 118 in age column. Also gender is a categorical data with possible values as F, M, and O. The length of membership also needs a conversion to number of days instead of anniversary date.

- **transcript.json**
  - event (str) - record description (ie 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

	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'}

Figure 3: Subset of Transcript data

Above shown transcript data represents the event log. An event log can either be offer received, offer viewed, offer completed or any other transaction. The dataset is used to prepare the training data set which includes the events when a customer is completed an offer, or a customer received an offer but never respond. Training data is taken from this dataset which includes true positive and true negative data. This data also need few cleaning such as conversion of categorical data even type, and normalize value column. Value column based on the event type may contain offer id, transaction amount, and reward amount.

Data Visualization

On visualizing customers' age, membership anniversary, and income distribution, it is found that most numbers of customers enrolled in the reward program are between 50–70 years of age. Also, there some record with incomplete data such as age is encoded 118 with no income information.

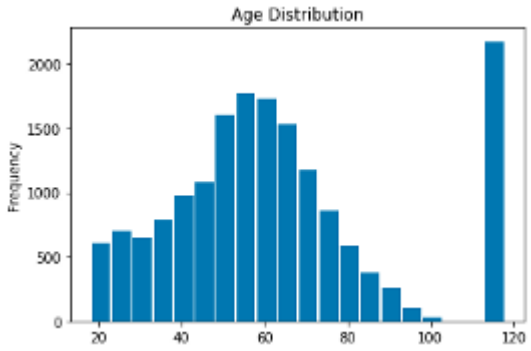

Figure 5: Age distribution

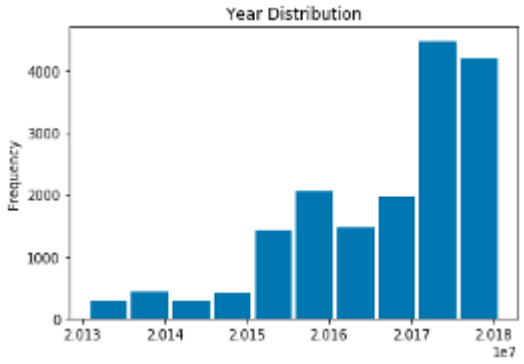

Figure 4: Membership year distribution

It is noticeable that Starbucks reward membership increased over the time since year 2013. But most customers signed up for reward program from 2017 to 2018. From income visualization, we can see that

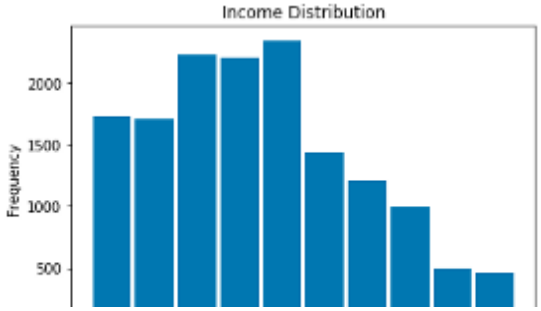

Figure 6: Income Distribution

majority of Starbucks reward program members have income range from 50,000 to 70,000 per annum.

Algorithms and Techniques

This is a supervised classifier problem with only two categorical label. Models for a given set of features of a customer's demographic along with offer details predicts whether or not the customer would accept the offer. So I consider DecisionTreeClassifier from scikit-learn package. As decision tree algorithm train model to generate a tree considering all the features in the training data and then predict the label, DecistionTreeClassifier seems to be a good candidate in this case. I will train the model with default parameter.

Benchmark

For comparing the performance of the above trained DecisionTreeClassifier I consider support vector classifier from scikit-learn package. Prediction result from

SVC will then be compared with the previous one. Both of the models perform considerably well given a relatively small dataset.

Methodology

Data Cleaning

The data cleanup is done in notebook includes following steps for each dataset.

- Portfolio dataset
  - Applying one-hot encoding to *channels* column.
  - Applying one-hot encoding to *offer_type* column.
  - Dropping *offer_type* column from.
- Profile dataset
  - Dropping record with missing age. In this case, missing *age* is encoded as 118. This in turn cleans missing income records.
  - Calculating membership days from *become_member_on* and dropping *become_member_on* column.
  - Applying one-hot encoding to gender column. In this case, genders are *male*, *female* and *other*.
  - Dropping *gender* column.

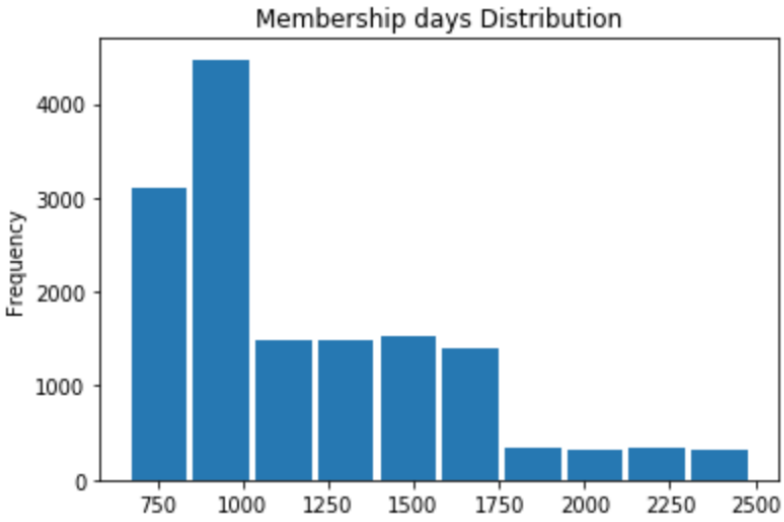

Figure 7: Membership days distribution

Above figure visualizes the membership days distribution. It shows the trend of customer enrolling to the reward program is increased recent days. That

justifies the correctness of the data conversion showing similarity with *become_member_on* year distribution.

- Transcript dataset
  - Extracting *offer_id* parsing dictionary string in value column.
  - Extracting *amount* parsing dictionary string in value column for event type transaction.
  - Creating new dataframe with *event_type*, *person_id*, *offer_id*, *amount*, and *reward*.

Data Preprocessing

The training dataset includes the features that I want the predictive model to be trained on. For this experiment, both customers' demographics and offer types of details are considered. So, the training data set is a new dataframe merged from Portfolio, Profile, and Transcript dataset. For true positive, a subset of the Transcript dataset with event type *offer_completed* is used. But for true negative, we will consider Transcript records with event type *offer_viewed* which customers do not participate in any transaction and there is no *offer_completed* event. Customers those who view offer but do not respond to the offer instead purchase Starbucks product are ignored from the training set.

After determining the Transcript subset applying above assumption, and then merging with Portfolio dataframe and Profile dataframe, the final training dataset looks like below:

```
Index(['label', 'age', 'income', 'membership_days', 'gender_male',
       'gender_female', 'gender_other', 'difficulty', 'duration', 'reward',
       'channel_web', 'channel_email', 'channel_mobile', 'channel_social',
       'bogo_offer', 'discount_offer', 'informational_offer'],
      dtype='object')
```

Dataset is then shuffled and split in training and testing dataset with 70% training data and 30% test data. I used scikit-learn convenient utility for this process.

Implementation

This is a binary classification problem where for a given customer demographic and offer features model tries to predict if the customer will accept the offer or not.

I use DecisionTreeClassifier and SVC models from scikit-learn package. Models are trained and are used to predict test dataset.

Accuracy of the prediction is measured with the help of scikit-learn *accuracy* util. Below is the accuracy of both the model. Overall accuracies of DecissionTreeClassifier and SVC are 0.98 (approx.) and 0.97 (approx.) respectfully.

Detail classification record of DecisionTreeClassifier and SVC is given below:

	precision	recall	f1-score	support
0	0.67	0.75	0.71	349
1	0.99	0.99	0.99	9725
avg / total	0.98	0.98	0.98	10074

Figure 9: Classification Report - Decision Tree

	precision	recall	f1-score	support
0	0.97	0.24	0.38	349
1	0.97	1.00	0.99	9725
avg / total	0.97	0.97	0.97	10074

Figure 8: Classification Report - SVC

Train model inputs are as below:

- Customer Details:
  - Age
  - Income
  - Number of days member of the reward program
  - Gender - male, female, or other
- Offer Details:
  - Difficulty - money required to be spent to receive reward
  - Duration – how long the reward valid
  - Reward – monetary amount
  - Channel – either email, web, mobile, or social
  - Offer Type – either bogo, discount, or informational

Refinement

Though the stat of training performance is good enough but there is one improvement can be done. The membership days distribution in Figure 7 shows that the number of membership has increased in recent days in other word there are more customer with less days as member. This distribution seems to be skewed which can be scaled using scikit-learn's StandardScaler and then train the models with the scaled data. After training with scaled data, I did not notice a significant improvement than what have so far.

	precision	recall	f1-score	support
0	0.69	0.75	0.72	349
1	0.99	0.99	0.99	9725
avg / total	0.98	0.98	0.98	10074

Figure 11: Classification Report - Decision Tree

	precision	recall	f1-score	support
0	1.00	0.20	0.33	349
1	0.97	1.00	0.99	9725
avg / total	0.97	0.97	0.96	10074

Figure 10: Classification Report - SVC

Though both models are performing well enough to be able to predict predict if a customer will accept for a given offer, the DecisionTreeClassifier is statistically doing better overall.

Conclusion

Free-From Visualization

Starbucks customers' age is almost a normal distribution. This leads a question on study more based on customers' age group.

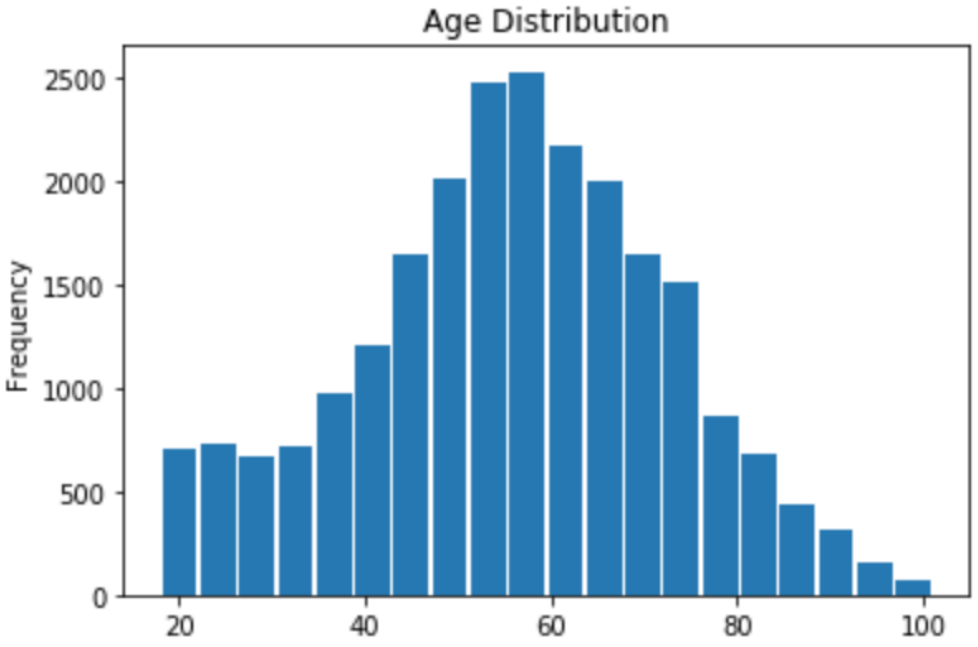

Figure 12: Cleaned up age distribution

An analysis on customers' age group and an overall correlation of their gender and income can be potentially identify the trends of their consumption of caffeine product. This can be done population segmentation using k-mean clustering. Another aspect of this dataset is that it can be used to infer the likelihood of consumption of a type of offer. This can be easily done training a classification model for a given set of offer type which in this case is 10.

Reflection

This project experiments the influence of customers' demographic and offers details on the acceptance of Starbucks rewards. The process start with analyzing the data. The data analysis process shows us the points where data needs cleaning. After cleaning data, I prepare a training dataset and preprocess to prepare the final training and test data. The challenging part of the project is cleaning the data to make data actionable. I train two classifier models with the training dataset. I use both the trained model to infer the likelihood of a customer's response on a given reward offer feature.

Improvement

There are few things can be improved. I think the model can be improved with more training data. After all the analysis and preprocessing, the training dataset seems to be inadequate. In addition, there might be room to improve training data quality by analyzing further details.