Capstone Project

Machine Learning Engineer Nanodegree

Lujing Yang
September, 2020

Starbuck's Project

1. Definition

1.1 Project Overview

The project focus on the customers response for the different types of Starbuck's offers. The datasets include the customer's profile, offer's information and the actions of the customers. The objective is to establish a model to predict a specific customer's response to a type of offer.

1.2 Problem Statement

In this project, I cleaned and transformed the datasets firstly. Then classified and analysed the customer clusters. Finally, a set of machine learning models were developed and compared. The specific steps are following:

1. Detecting and clean the datasets of portfolio, profile and transcript;
2. Transforming the datasets;
3. Clustering the customers and analysing the response of different groups;
4. Developing benchmarking model to predict the possibility of customers using an offer;
5. Transferring the problem to classification problem;
6. Developing different classification models and comparing.

1.3 Metrics

Accuracy, Precision and Recall are common metrics for binary classifiers. Accuracy is calculated by both true positives and true negatives with equal weight:

$$Accuracy = \frac{\text{True Positives } + \text{ True Negatives}}{\text{sample size}}$$

This metric is used to evaluate the overall score of models.

- ✓ True positive means the model predicts successfully a customer will use or view an offer, and the customer really use this offer.
- ✓ True negative means the model predicts a customer will NOT use or view an offer, and the customer really does not use or view it.

Precision is calculated by true positives and false positive:

$$Precision = \frac{\text{True Positives}}{\text{True Positives } + \text{ False Positives}}$$

Precision is a metric that is used to evaluate how many real positive samples in all samples predicted by a model. The metric is very important when the business needs the model should avoid making wrong positive prediction as much as possible.

- ✓ False positive means the model predicts a customer will response to an offer, but the customer does not actually.

Recall is calculated by true positives and false negatives:

$$Recall = \frac{\text{True Positives}}{\text{True Positives } + \text{ False Negatives}}$$

Recall is used to evaluate the model's ability to find the positive samples from dataset. It is very important when people need to find as much as possible samples they want.

- ✓ False negative in this project means a model predict a customer will not response to an offer, but the customer uses or views it.

2. Analysis

2.1 Data Transformation

The datasets include the offer's information (portfolio.json), customer's information (profile.json) and events (transcript.json). The data schema of them are following:

Field Name	Data Type	Definition
portfolio		
id	string	The unique id of an offer
offer_type	string	Type of offer including Bogo, Discount and Informational
difficulty	int	Minimum (steps?) required spend to complete an offer
reward	int	Reward given for completing an offer
duration	int	How many days for offer to be open
channels	List of string	The channel names for delivering offers to customers
profile		
age	int	Age of customer
became_member_on	number	Date when customer created an app account (yyyymmdd)
gender	string	Gender of the customer
id	string	Customer id
income	float	Customer's income
transcript		

event	string	Event description (transaction, offer received, offer viewed and offer complete.)
person	string	Customer id
time	int	Hours since start of test. The data begins at time t=0
value	dict of strings	Offer id or transaction amount depending on the record

According to the above table, many fields in these three datasets are string of other types containing strings. To build a machine learning model, it is better to transform them to numeric fields. Moreover, the datasets also may need clean because of NA or abnormal values.

2.1.1 Portfolio:

For the dataset of portfolio, it does not have any NA or abnormal values, so it does not need to be cleaned.

The field of channels is a list of string, to establish classification models, it should be transformed to numeric field. I created four new fields to replace channels, which are channel_email, channel_mobile, channel_social and channel_web. These fields will be 1 if the original channels field contains the channel name, or it will be 0. For example, if an offer's channel field is [social, web, mobile], the new fields channel_mobile , channel_social and channel_web will be 1 and channel_email is 0.

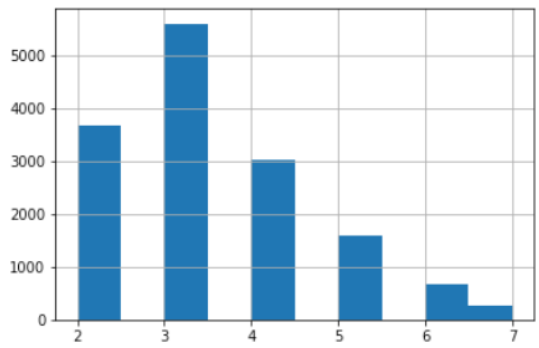
2.2.2 Profile:

This dataset includes 2175 rows containing NA values of the fields of income and gender, and all the rows have an abnormal age value of 118. After deleting these rows, the dataset does not have any NA values and abnormal value.

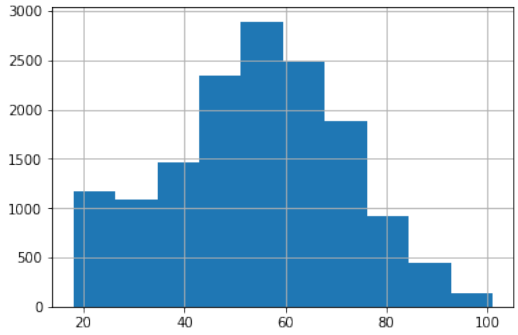
In next, some fields of this dataset need to transform for modelling:

a) The gender field contains values of 'M' (8484), 'F' (6129) and 'O' (212). Transforming the field to is_male, which is 1 if gender is 'M', or is 0.

b) The year when a customer become a member is transform to number age (year), which is more efficient to represent the membership duration. After that, the number age distribution is following:

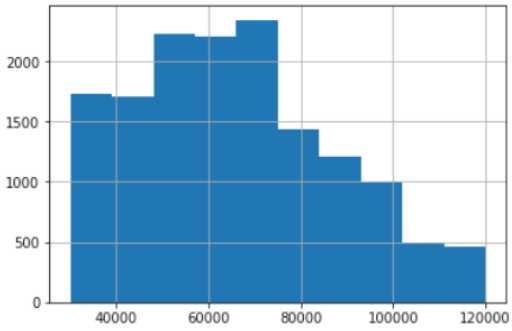


c) The age field's distribution is following:



It likes a normal distribution which mean is around 50 year. This field is transform to age-stage by mode 10.

d) The situation of customers' income is shown in here:



Similar to age, the income is also transformed to different levels every 10,000.

2.1.3 Transcript:

The dataset of transcript is a little complex for transformation. First, offer id and transaction amount are extracted from 'value' field. Then the count of every type of event for a customer and an offer are calculated to generate a new dataset with the records like these:

Person	Offer_id	event	count
0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5	offer completed	1
0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5	offer received	2
0009655768c64bdeb2e877511632db8f	2906b810c7d4411798c6938adc9daaa5	offer viewed	2

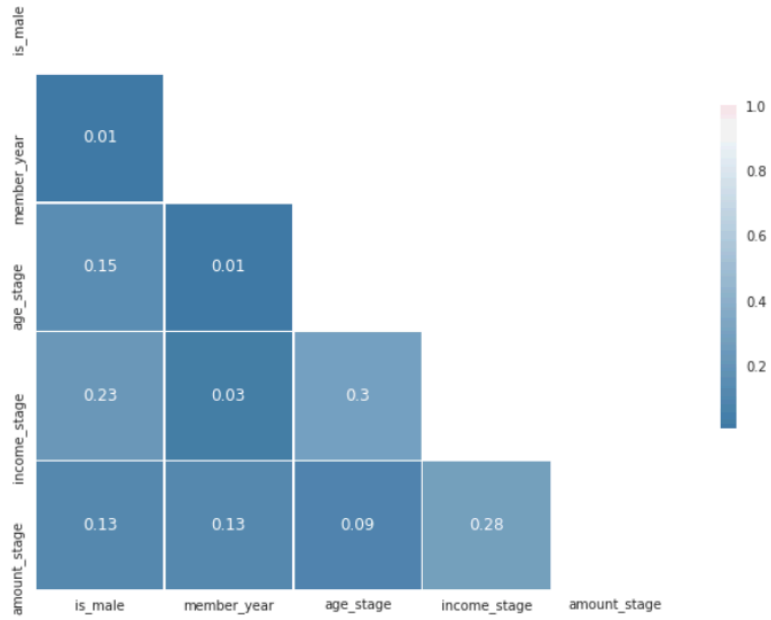
The above records the person (id is '0009655768c64bdeb2e877511632db8f') received the offer (id is '2906b810c7d4411798c6938adc9daaa5') 2 times, review 2 times and use it 1 time.

Then the new dataset is merged with profile (customers' data) and portfolio (offers' data) to generate the training dataset for modelling.

2.2 Customer Clustering

In order to analyse different type customers' response to different type of offers, we try to cluster customers by using K-means algorithm.

Before clustering, the profile dataset fields' values need to be scaled to avoid the overweighting the large scale feature. MinMaxScaler of sklearn is used here and the correlation matrix between fields of scaled dataset is following:



The corr-matrix shows the features have high independence for others so they can be used in K-means directly. Three clusters are generated and each cluster's features are displayed in following table:

cluster_type	is_male	member_year	age_stage	income_stage	amount	amount_stage
0	0.663378	2.553527	4.352574	4.770926	73.132204	0.390561
1	0.415667	3.236211	6.038969	8.407274	162.647264	1.150879
2	0.633690	4.915508	4.625401	5.189572	127.363251	0.790374

From the table above, the type 0 has more female customers, their age and income are higher, and spend more money to buy from Starbucks; the type 1 and type 2 customers are younger and have lower income. But type 1 customer has longer member age than other types, and this type of customers spend much more than type 2.

These three clusters of customers also have different responses for each type of offers. The counts of different events of different customer clusters response to each type of offer are calculated as following:

offer_type	cluster_type	offer_received	offer_viewed	offer_completed
bogo	0	10808	8939.0	4479.0
	1	8967	7368.0	6453.0
	2	6762	5732.0	4326.0
discount	0	11056	7308.0	5189.0
	1	8911	6509.0	6789.0
	2	6697	4644.0	5208.0
informational	0	5475	3756.0	0.0
	1	4472	3135.0	0.0
	2	3353	2469.0	0.0

From the table above, type 0 and 1 customers have high possibility level to response offer. Type 2 customers are not like to response to offers. For the customers of Type 1, they have more higher rate of using discount offer than using Bogo offer.

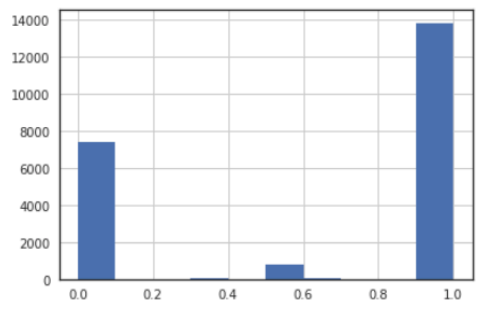
3. Modelling

3.1 Data Preprocessing

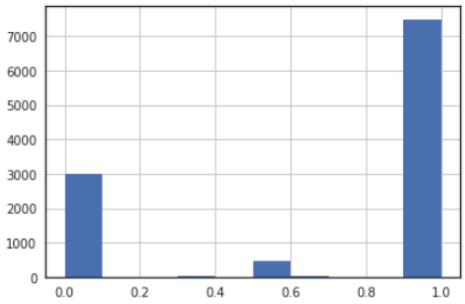
a) Dividing Dataset: Because dataset has three types of offers, and different type will lead to different actions. For example, informational offers cannot be used. Therefore, we needs to develop models for these three types of offer separately. It requires the whole dataset should be divided into three sets based on the offer type.

b) Generating Rate of Response: In next, we should generate the label of data. Firstly, a new field of completed rate can be calculated by using received count and completed count. For example, if a customer receive an offer 2 times but only use it 1 time, the completed rate is 1/2 = 0.5. For the dataset of informational offer, because it cannot be used so only viewed rate can be calculated.

The distribution of completed rate is shown in following diagram, which means in most cases, customers will response to all or never.



The viewed rate distribution is similar:



c) According to the polarised distribution of response, we can simply the response to binary classification label. In this project, if the rate is greater than 0.5, the new field 'is_viewed' or 'is_completed' is set as 1, else it will be 0.

d) Splitting the dataset into training data (80%) and testing data (20%).

3.2 Benchmarks (Regression Model to Predict Completed/Viewed Rate)

To predict the rates, the regression model is developed. Random Forest Regressor is chosen to develop the model. Grid Search of sklearn library is used to search the best parameters:

```
param_grid={'max_features': ['auto', 'sqrt'],
                'max_depth' : [10,15,25],
                'n_estimators': [10,20,25,30,45],
                'min_samples_split': [10,20],
                'min_samples_leaf': [10,15],
                'bootstrap':[True, False]
                }
grid_search_bogo = GridSearchCV(RandomForestRegressor(random_state=2), param_grid)
grid_search_bogo.fit(X_train, y_train)
grid_search_bogo.best_params_
```

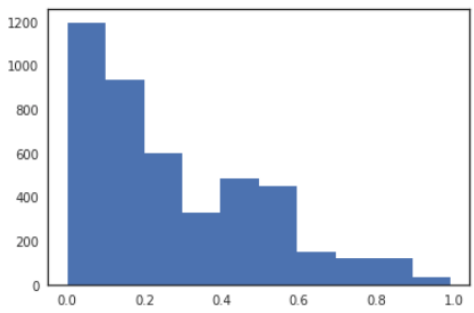
```
{'bootstrap': True,
 'max_depth': 10,
 'max_features': 'sqrt',
 'min_samples_leaf': 15,
 'min_samples_split': 10,
 'n_estimators': 45}
```

The errors of these three types of offers:

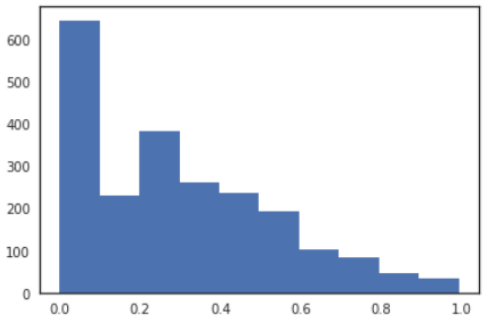
	Bogo Completed Rate	Discount Completed Rate	Informational Viewed Rate
Average Error	0.3	0.28	0.3

Because the rates are between 0 to 1.0, the errors are very high. The predict values' distributions also have big gap compared to real values:

Completed Rate Distribution:



Viewed Rate Distribution:



From the results of Random Forest Regressor models, we can find the regression model is not good for this problem. Therefore, we should try the way that transferring to classifier problem and using classifier model to predict.

3.3 Binary Classification Models

Many machine learning methodologies can be used to develop binary classification model. In this project, we choose Random Forest Classifier, Logistic Regression and SVM. GridSearchCV method is also used to search the best parameters. For example, the codes of training different models for Bogo offers are illustrated as this:

Random Forest Classifier:

```
param_grid={'max_features': ['auto', 'sqrt'],
                'max_depth' : [10,15],
                'n_estimators': [10,20,25,30],
                'min_samples_split': [10, 20],
                'min_samples_leaf': [10,15],
                }
grid_search_bogo = GridSearchCV(RandomForestClassifier(random_state=2), param_grid)
grid_search_bogo.fit(X_train, y_train)
grid_search_bogo.best_params_
```

```
{'max_depth': 10,
 'max_features': 'auto',
 'min_samples_leaf': 15,
 'min_samples_split': 10,
 'n_estimators': 20}
```

Logistic Regression:

```
params = {
            'penalty' : ['l1', 'l2'],
            'C' : np.logspace(-4, 4, 20),
            'solver' : ['liblinear'],
         }

grid_search_bogo = GridSearchCV(LogisticRegression(),params)
grid_search_bogo.fit(X_train, y_train)
grid_search_bogo.best_params_
```

```
{'C': 0.088586679041008226, 'penalty': 'l1', 'solver': 'liblinear'}
```

SVM:

```
params = {'C': [0.1,1], 'gamma': [0.1,0.01,0.001],'kernel': ['rbf', 'sigmoid']}
grid_search_bogo = GridSearchCV(SVC(),params, refit=True,verbose=2)
grid_search_bogo.fit(X_train, y_train)
grid_search_bogo.best_params_
```

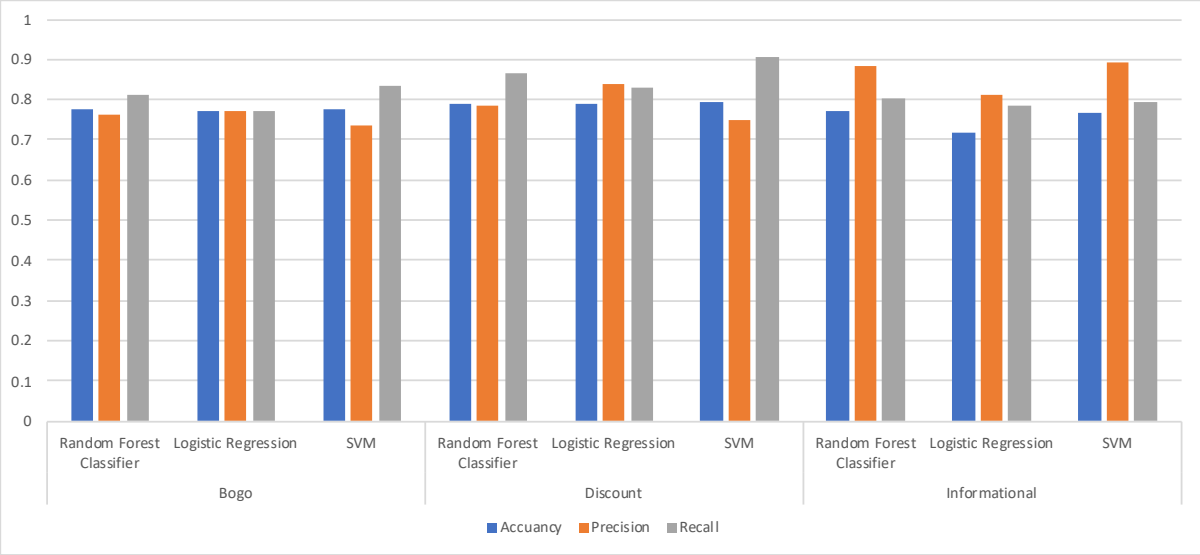
{'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}

The metrics of models' performance on test data are listed in following table:

Offer Type	Model	Accuracy	Precision	Recall
Bogo	Random Forest Classifier	0.775	0.761	0.81
	Logistic Regression	0.772	0.731	0.831
	SVM	0.775	0.736	0.833
Discount	Random Forest Classifier	0.79	0.786	0.867
	Logistic Regression	0.788	0.837	0.828
	SVM	0.792	0.749	0.904

Offer Type	Model	Accuracy	Precision	Recall
Informational	Random Forest Classifier	0.771	0.882	0.805
	Logistic Regression	0.717	0.812	0.785
	SVM	0.769	0.893	0.796

A bar chart is created to compare the metrics:



The performance of these three model do not have significant gap. Generally speaking, SVM has the best performance. Random Forest Classifier's performance is very closed to SVM and better than Logistic Regression.

Although the performance is a litter lower than SVM, Random Forest Classifier has potential to expand to a large scale model (I did not train a large forest since the system limitation), and has high toleration to non-linear features. And the training time is less than SVM. So I recommend Random Forest Classifier model.

4. Conclusion

This report discusses the methodologies of clean, transform, analysis of the Starbucks' datasets, and develops a set of machine learning models to predict customer's response to offers.

The job of datasets' clean in this project is simple, since the data does not contain too many NA and fault values. The datasets' transformation job is more complex. First, the original dataset of transcript and portfolio are not complete structured so some fields are parsed to a set of new features; second, the string-based features are transformed to numeric features; third, some fields with continuous values are transferred to concreted values to suitable for modelling; last, we generate some new features (include the target) by calculating others.

From the works of modelling, it is found that this project is not suitable to be solved as a regression problem. One possible reason is that the targets of data (viewed rate and completed rate) are concentrated at 0 and 1. This type of distribution is more suitable for classification rather than regression.

Therefore, we choose the way to transfer the problem to a binary classification problem. The models just need to predict wether a customer responses to an offer. From the results of metrics for each models, SVM has the best performance on current datasets but Random Forest Classifier has more potential to be used in real world.

There are some points for improvement. First of all, the data of events also includes the value of 'offer viewed' for Bogo and Discount offers. In current work, we just consider if a customer uses a Bogo or Discount offer finally because it is the objective of these offers. In future, if the business focuses on how to promote the conversion rate from offer viewing to completing, the models based on response of 'offer viewed' also need to be developed.

Secondly, the structure of Random Forest Classifier can be enlarged by using more estimators and depth level if we can get more data and have more computing resource. Moreover, the Logistic Regression and SVM models can have more search scope for their hyper parameters.

The third point is that we can pay more attention on customer clustering. In current work, the customers a clustered into three groups. We can try to cluster them into 2 or 4 groups and investigate the results.

Last but not least, the offer channel analysis also can be analysed, which is more detailed than offer type. The business can find the performance of each channel and decide what is the best way to delivery offer to customers.