Capstone Project

Natthapong Chantaklad

Data science Nanodegree

February 16th, 2021

Definition

Project Overview

One of the main problems that the company faces when sends out an offer to the customers is the company doesn't know which offer be preferred by customers. Hence the company sends out an offer without knowing what offer that customer desires.

This project predicts whether offers is responded or not responded by an individual customer. This means that the company can implement this prediction to help the company find which offer is preferred by an individual customer and sends out an offer correctly. The project implements the XGBoost classifier and Starbucks dataset to training the model.

The project implements the XGBoost classifier and Starbucks dataset. The Starbucks dataset comprised Portfolio, Profile, and Transcript dataset. These datasets are used to train the XGBoost model.

Problem Statement

The objective is to find which offers is responded by an individual customer. The processes are the following.

1. Clean the Starbucks datasets (portfolio, profile, and transcript).
2. Create the dataset that corresponds to the objective.
3. Clean the objective dataset.
4. Train and test the XGBoost classifier.
5. Predict whether an offer is responded by an individual customer or not by using the trained XGBoost classifier.

Metrics

Accuracy is a metric that define whether the model good or not for prediction. This metric is commonly use with binary classifier.

$$Accuracy = \frac{True\ positives + True\ negatives}{The\ number\ of\ observations}$$

True positive
The positive values that are correctly classified.

True negative
The negative values that are incorrectly classified.

Analysis

Data Exploration

Starbucks dataset consist of portfolio, profile, and transcript dataset. The datasets represent offer portfolio, customer profile, and customer behavior on the Starbucks rewards mobile app.

Once every few days, Starbucks sends out an offer to customers on the mobile applications. An offer can be a BOGO (buy one get one free) or a discount or just an advertisement/information for a drink. Some customers might not receive any offers during certain weeks.

Portfolio
The portfolio dataset represents the data of each Starbucks offer. The data structure and explanation of each variable in the portfolio dataset are as follow:

- id (string) – offer id
- offer_type (string) – type of offer; either BOGO, discount or informational
- difficulty (int) – minimum dollars that require to spend to get an offer
- reward (int) – reward that customer received after completed an offer
- duration (int) – time that an offer available for completing the offer
- channels (list of strings) – list of channels that can receive an offer (web, email, mobile, social)

Profile
The profile dataset represents the demographic data of each customer. The data structure and explanation of each variable in the profile dataset are as follow:

- age (int) – customer age
- became_member_on (int) – the date when a customer created an application account
- gender (string) – customer gender; either M, F, or O. The 'O' value is for other genders
- id (string) – customer id
- income (float) – income from customer

Transcript
The transcript dataset represents the customer behavior on the Starbucks rewards mobile app. The data structure and explanation of each variable in the transcript dataset are as follow:

- event (string) – record description; either transaction, offer received, offer viewed or offer completed
- person (string) – customer id
- time (int) – time in hours since the record begins. The time initial value is 0
- value (dict of strings) – value consists of three unique variables; offer id, amount and transaction. Which differently assign in each record depending on an offer event

An example of transcript situation

The company sends an offer to a customer via different channels depending on an offer. Then the customer received an offer. The offer has a duration time and a condition to complete. For instance, the company sends a discount buy 10 dollars get 5 off to a customer via website and email. Then the custom received the discount offer on Monday. The offer is valid for 7 days and has a condition to complete the offer.

The customer needs to accumulate at least 20 dollars in purchases during the valid period to get an offer. If the customer accumulative spend meet 10 dollars and complete the offer in time, the customer completes the offer and receives a 5 dollars discount.

There is cautious information about the datasets: A customer might complete an offer without viewing the offer or received an offer but never open the offer during the available period.

Exploratory Visualization

Portfolio dataset

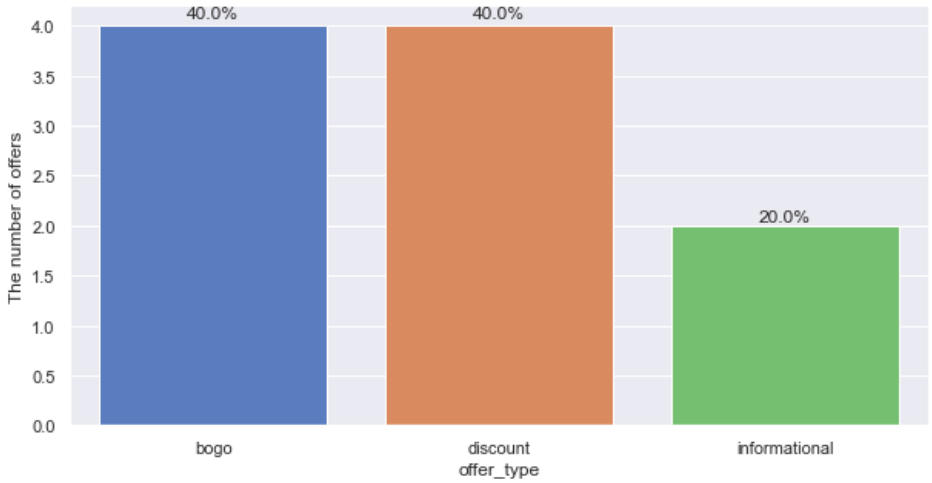

Fig. 1 The number of each offer type

The bar chart above shows the number of each offer type. It is clear that BOGO and Discount are equal in the number of offers at 40% followed by informational with 20%, respectively.

This information also shows the proportion of offers that a company sends out to an individual customer.

Profile dataset

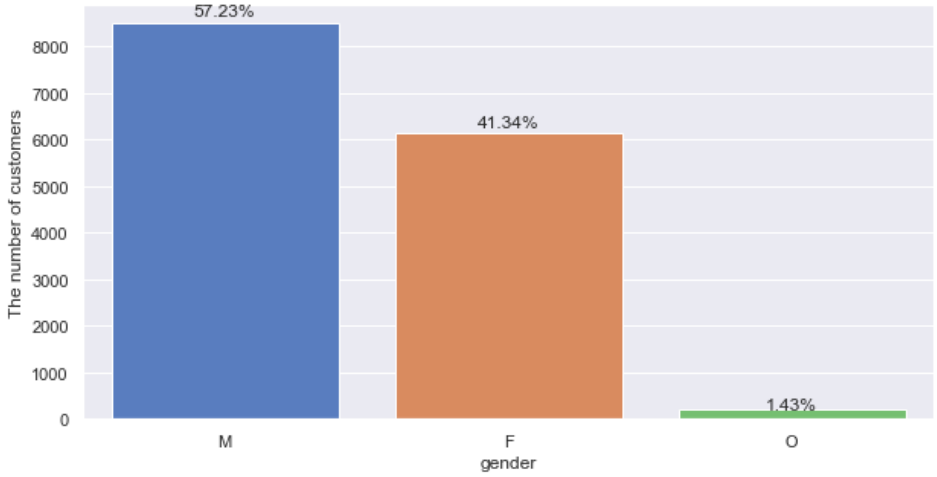

Fig. 2 The number of each gender

The bar chart above illustrates the number of each gender. It clearly sees that Male is the most group of customers at 57.23% followed by Female 41.34% and Others 1.43%, respectively.

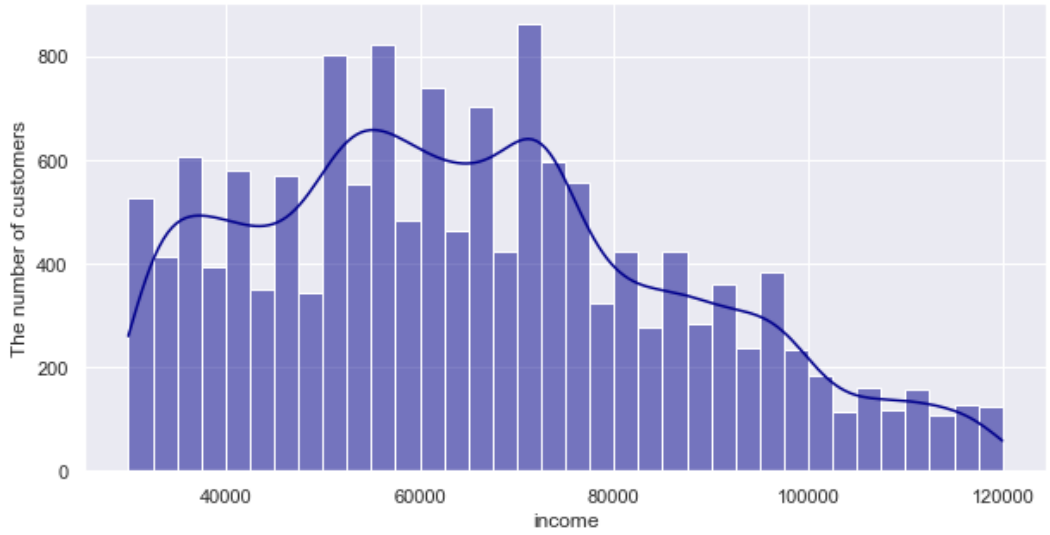

Fig.3 The distribution of customer income

The plot above illustrates the distribution of customer income. It clearly sees that most income are between 50000 and 75000 dollars with the 3000 at minimum and 12000 at maximum income in dollars.

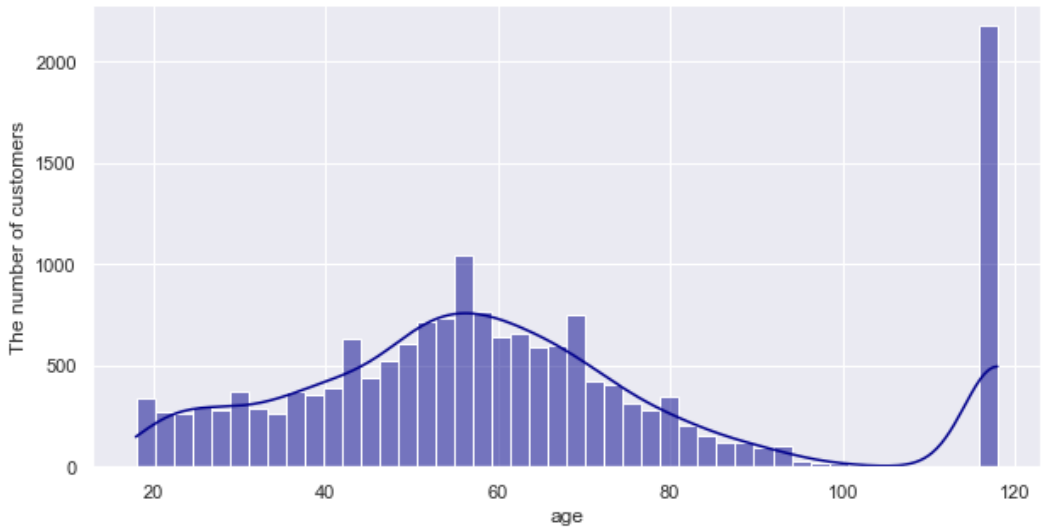

Fig. 4 The distribution of customer age

The distribution plot above shows the distribution of customer age. At first glance, there is an abnormal age value or an outlier at 118 years old. This outlier will be ignored or removed when describes the distribution.

The distribution without the outlier can be described as follow. Most customer ages are around 50 and 65 years, with 18 at minimum and 101 at maximum.

The table below is showed to investigate why the abnormal value is presented in the customer age distribution.

	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN

Fig. 5 The profile dataset query by abnormal age (118)

The table above shows that the abnormal age value (118) has a relationship with the variable that contains NaN/None values (missing values). According to the relationship, the 118 Age value can be considered as a missing value because when the Gender and Income went missing, the Age value always at 118 years old.

The Age values are set to 118, maybe because whenever the Age doesn't have a value, the company sets an age value to the maximum (118).

To investigate further about this relationship, the missing value heatmap is showed below.

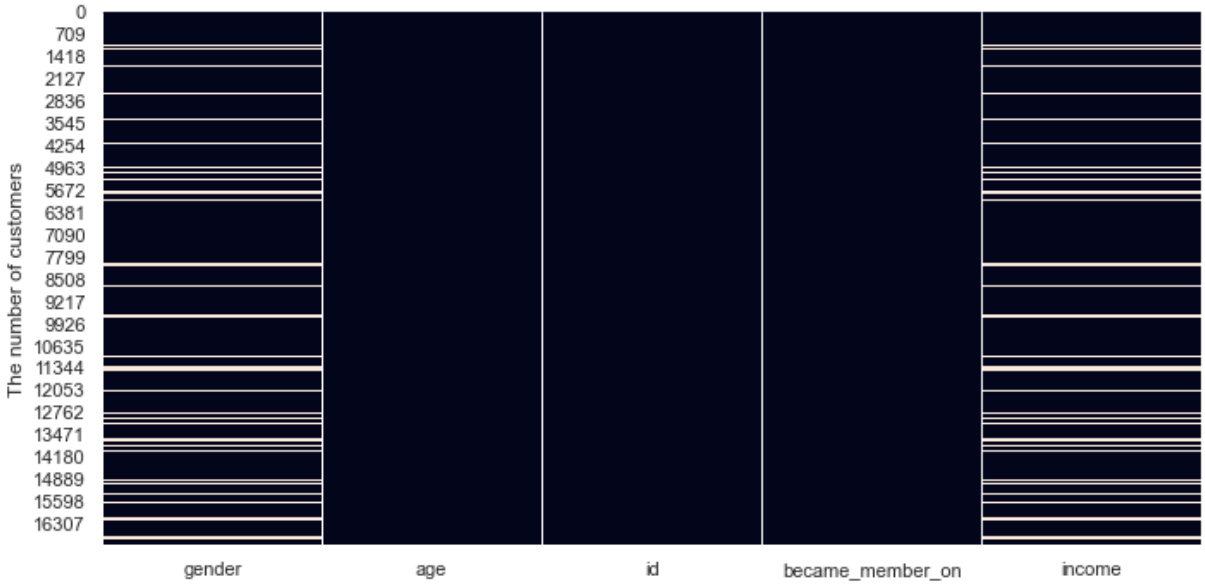

Fig. 6 The missing value heatmap of Profile dataset

The plot above shows the missing value distribution across variables. It can clearly see that Gender and Income have similar missing value patterns. This means that Gender, Income, and Age values have gone missing for the same reason.

The reason why these variables went missing maybe is that a customer doesn't give information to the company or just register an account and don’t purchase anything. Note that the missing values are handled in the data preparation part.

Transcript

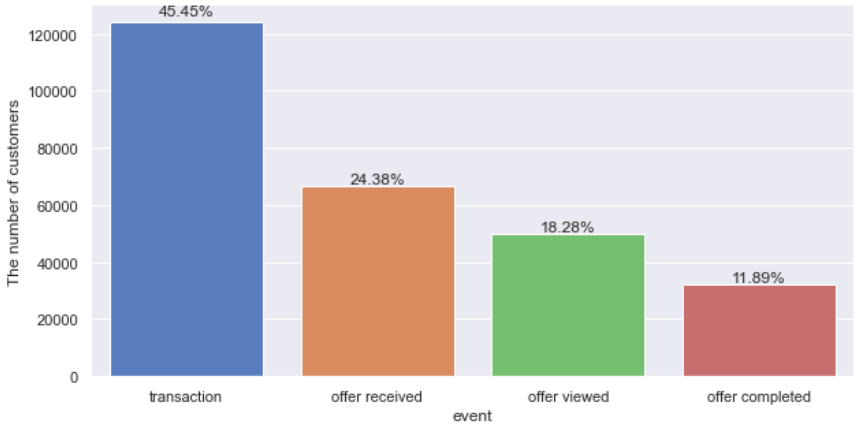

Fig. 7 The number of each event type

The bar chart above illustrates the number of each event type. It clearly sees that transaction is the most type of event at 45.45% while offer received, offer viewed, and offer completed are 21.38%, 18.28%, and 11.89%, respectively.

Algorithms and Techniques

The classifier is an XGBoost, which is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way[1].

The steps of building the XGboost classifier are as follow:

1. Define the initial prediction for the first prediction. The default initial predicted value is 0.5 for the classifier.
2. Find the residuals by subtracting the target values from the previous prediction value. Note that for the first prediction, the initial prediction is used to subtract the residuals.
3. Building an XGBoost tree from the residuals.

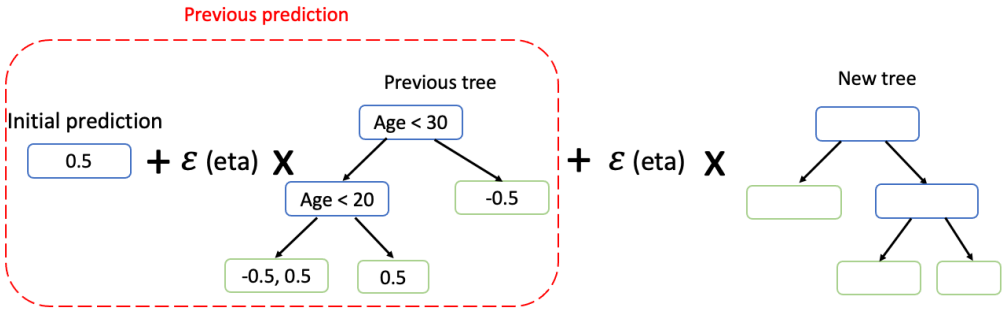

Fig. 8 An example of the previous prediction

The XGBoost trees are built as follow:

$$Loss\ function: L(y_i, y_p) = \left[\sum_{i=1}^{n} L(y_i, p_i^0 + O_{value})\right] + \frac{1}{2}\lambda O_{value}^2$$

The regression loss function (L2 Regularization) for finding optimal value equation

$$O_{value} = \frac{-(g_1 + g_2 + \cdots + g_n)}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

The optimal value equation

$$Loss\ function: L(y_i, y_p) = -[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

The classification loss function equation

The goal is to minimize the loss function.

---

[1] XGBoost Documentation <https://xgboost.readthedocs.io/en/latest/> (2020).

Do the math of the equations above, the result is the Similarity Score equation below.

$$Similarity\ Score = \frac{(\sum Residual_n)^2}{\sum[Previous\ Probability_n\ x\ (1 - Previous\ Probability_n)] + \lambda}$$

Note that for the first iteration, the Previous Probability is the initial predicted value. And λ is the regularization parameter that is used to reduce the prediction's sensitivity to isolated observations. In other words, lambda (λ) helps to prevent overfitting the training data by the individual observations.

The steps of building the XGBoost tree are as follow:

1.1 Choose candidates among observations to split residuals (the candidates are selected by the average of two observations that close to each other).
1.2 Create the branch of residual by each candidate.

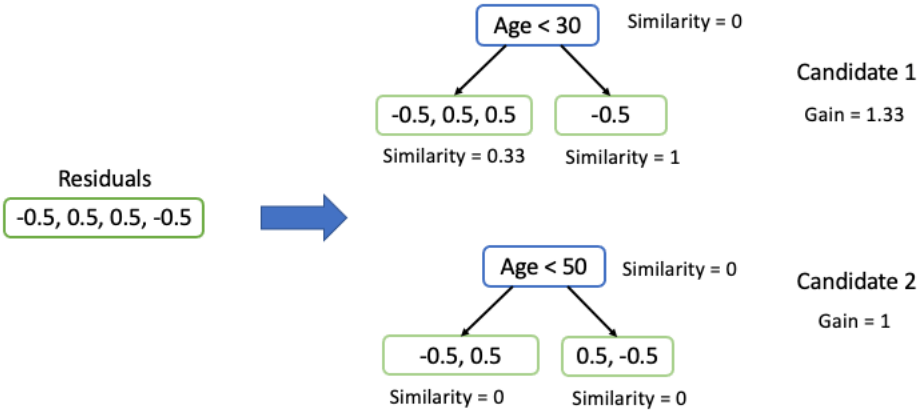

Fig. 9 An example of how XGBoost classifier splits the residuals

1.3 Calculate the similarity score of each candidate branch.
1.4 Calculate the Gain of each candidate branch.

$$Gain = \sum Leaf\ Similarity_n - Root\ Similarity$$

Note that the similarity scores equation is from the candidate branch, not the entire tree.

1.5 The candidate branch that has the most Gain is selected to be a part of the tree.
1.6 The classifier keeps split the residuals until the tree depth is reached.
1.7 After finished building the tree, the classifier starts pruning the tree by subtracting the Gain of each branch by γ (gamma):

$$Gain - \gamma$$

If the difference between the Gain and γ is negative, the branch is removed.

According to the equations, the higher λ value the lower the Gain value. In other words, the higher λ the easier to prune the tree.

1.8 After pruned the tree, the new tree is finished.

4. After successfully built the new tree, to get the new prediction, the initial prediction or previous prediction values need to be converted to log(odds) values as the following equation:

$$\log(odds) = \log\left(\frac{p}{1-p}\right)$$

5. Then convert to probability.

$$\text{Probabilty} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

6. And multiplied by the learning (eta).

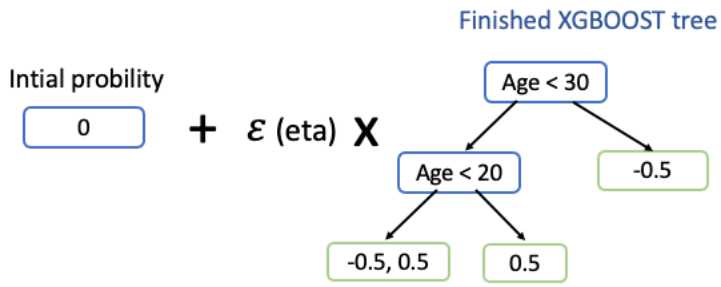

Fig. 10 An example of getting a new prediction

7. Then add the converted value with the prediction from the new tree.
8. The XGBoost Classifier keeps doing the above steps until the residuals are super small or reach the maximum number.

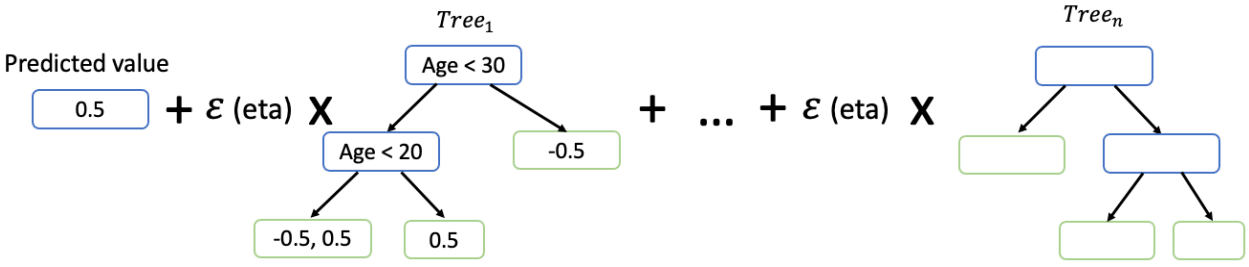

Fig. 11 An example of Building an XGBoost classifier tree process

The following hyperparameters are tuned to optimize the classifier.

- Eta – learning rate
- Gamma – The value that use to prune the tree
- max_depth – Max tree depth
- Lambda - L2 regularization
- min_child_weight - Cover

$$Cover = \sum [Previous\ Probability_n\ x\ (1 -\ Previous\ Probability_n)]$$

The cover (min_child_weight) is a threshold for the minimum number of residuals in each leaf.

- eval_metric – an evaluation metric (Benchmark)
- early_stopping_rounds – stop a validation if the eval_metric score doesn't improve for n rounds

Benchmark

To measure a classifier benchmark, The Area Under the Curve (AUC) of the ROC graph is used. The measurement operates by comparing the AUC of each XGBoost classifier iteration (the classifier model changes every iteration). The classifier model that has the most AUC, is considered as the best classifier.

Receiver Operator Characteristic (ROC)

The ROC graph shows a diagnostic ability of a binary classifier by comparing each classifier threshold. The ROC graph compares the True positive rate and False positive rate values in percentage.

True positive rate or Sensitivity

The proportion of the positive values that are correctly classified.

False positive rate

The proportion of the negative values that are incorrectly classified.

For instance, if the true positive rate is 0.8 and False positive rate is 0.3, a classifier correctly labels the 80% of positive values and incorrectly labels the 30% of positive values.

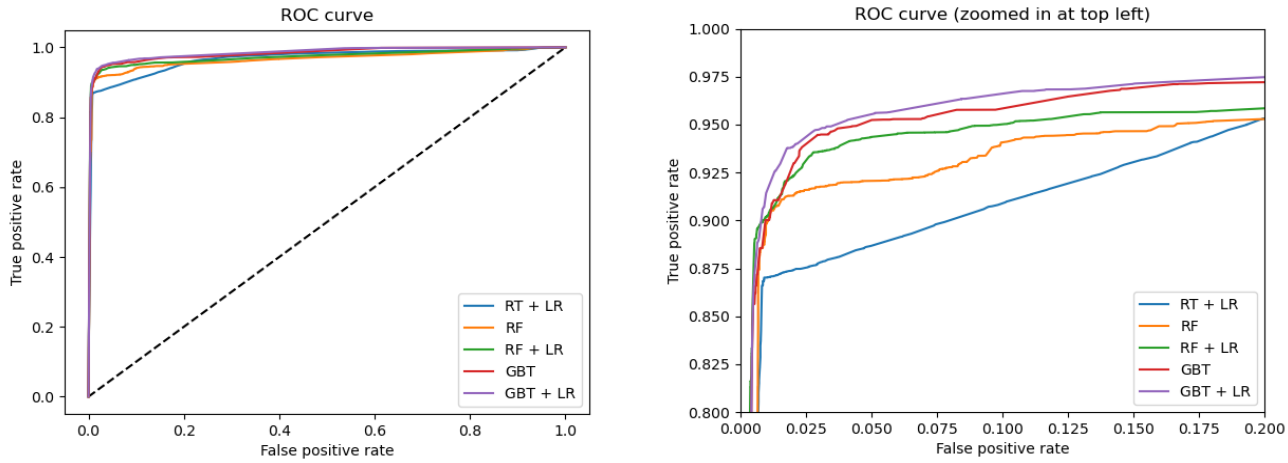

Fig 12. An example of a ROC curve

The plots above show the comparison of each classifier. The classifier that has the most AUC will choose as the best classifier. For instance, GradientBoosting classifier with Logistic regression (GBT+LR) has the most Area Under the curve (AUC) in the ROC graph then the GBT+LR will be chosen as the best classifier.

Methodology

Data Preprocessing

The data preprocessing or cleaning done by completing the steps below:

1. The portfolio dataset cleaning process

	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

Fig. 13 An example of records in portfolio dataset.

1.1 Find missing values for each column.

Find the columns that contain missing values by calculating the mean of missing values. But the missing values are not found in this dataset. Therefore, the drop missing values method is not used in this dataset.

1.2 De-hash id to readable id

Convert hashed id to readable id to reduce computation time.

hash_offer_id	offer_id
ae264e3637204a6fb9bb56bc8210ddfd	1
4d5c57ea9a6940dd891ad53e9dbe8da0	2
3f207df678b143eea3cee63160fa8bed	3

Fig. 14 An example of convert hashed id to readable id

1.3 Create one-hot encoding for channels column

Convert categorical values to numeric values to be able to put in the classifier.

channels	email	mobile	social	web
[email, mobile, social]	1	1	1	0
[web, email, mobile, social]	1	1	1	1
[web, email, mobile]	1	1	0	1
[web, email, mobile]	1	1	0	1
[web, email]	1	0	0	1

Fig. 15 An example of one-hot encoding

1.4 Rename columns

To make columns more readable ex. Id -> hashed id.

2 The profile dataset cleaning process

	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

Fig. 16 An example of records in profile dataset.

2.1 Find missing values for each column.

Find missing values by calculating the mean of missing values for each column.

2.2 Investigate missing values

Investigate a missing values relationship. After investigation, an abnormal relationship is found between missing values and variables according to the data exploration section above.

2.3 Drop missing values

Drop the missing values that have an abnormal relationship and don’t have a relationship.

2.4 Convert date string to date type

Convert likely date string to date type.

2.5 De-hash id to readable id

Convert hashed id to readable id to reduce computation time.

2.6 Rename columns

To make columns more readable.

3. The transcript dataset cleaning process

	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

Fig. 17 An example of records in transcript dataset.

3.1 Find missing values for each column.

Find the columns that contain missing values by calculating the mean of missing values. But the missing values are not found in this dataset. Therefore, the drop missing values method is not used in this dataset.

3.2 Extract variables from value column

Extract the hashed offer id, reward, and amount from dictionary value columns to new columns.

value
{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}
{'offer id': '3f207df678b143eea3cee63160fa8bed'}
{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}

hash_offer_id	reward	amount
9b98b8c7a33c4b65b9aebfe6a799e6d9	NaN	NaN
2906b810c7d4411798c6938adc9daaa5	NaN	NaN
f19421c1d4aa40978ebb69ca19b0e20d	NaN	NaN
3f207df678b143eea3cee63160fa8bed	NaN	NaN
0b1e1539f2cc45b7b9fa7c272da2e1d7	NaN	NaN

Fig. 18 An example of the extracting variables from value columns

3.3 De-hash id to readable id

Convert hashed id to readable id to reduce computation time.

3.4 Drop the dehashed customer ids that have missing values

The missing value occurs because this dataset contains abnormal customer ids likewise the profile dataset. Hence, the missing value must be removed according to the profile dataset.

3.5 Rename columns

To make columns more readable.

4. Build a training dataset

Build training dataset from three datasets from cleaned datasets above.

4.1 Select non-transaction events (offer events)

Separate the non-transaction events from transaction events.

4.2 Create the dictionaries by following criteria:

- responsed_dict: The dictionary between customer_id and the list of responsed offer
- offer_dict: The dictionary between customer_id and the list of viewed offer

4.3 Build a training dataset from the portfolio, profile, and offer transcript datasets by using the dictionaries above. The target values (response) are created by the following conditions:

- response = 0; an offer is responded
- response = 1; an offer is not responded

	gender	age	hash_customer_id	became_member_on	income	customer_id	response	reward	channels	difficulty	duration	offer_type	hash_offer_id	offer_id	email	mobile	social	web
0	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	1	0.0	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	4	1	1	0	1
1	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	1	0.0	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed	3	1	1	0	1
2	F	75	78afa995795e4d85b5d9ceeca43f5fef	2017-05-09	100000.0	2	1.0	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	4	1	1	0	1
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	2017-05-09	100000.0	2	0.0	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837	8	1	1	1	0
4	F	75	78afa995795e4d85b5d9ceeca43f5fef	2017-05-09	100000.0	2	1.0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1	1	1	1	0

Fig. 19 An example of training dataset

5. Clean the training dataset

5.1 Drop unused columns. The reasons are as following:

- channels: Duplicates with hashed offer id, reward, and amount columns
- hash_offer_id: Duplicates hashed with offer_id id column
- id columns: Not provide a meaningful analysis
- became_member_on: Doesn't use to analysis

5.2 Find missing values for each column

Find the columns that contain missing values by calculating the mean of missing values. But the missing values are not found in this dataset. Therefore, the drop missing values method is not used in this dataset.

5.3 Remove outliers

An outlier is a crucial problem that makes a model inefficient. It should be handled before performing any actions that relate to machine learning or modeling. The outlier in Age and Income columns are removed by the following methods:

The statistic z-score

Filter outliner by statistic z score. The values that have a z-score > 3 are dropped.

Isolation Forest

Isolation Forest is an outlier detection technique that identifies anomalies instead of normal observations

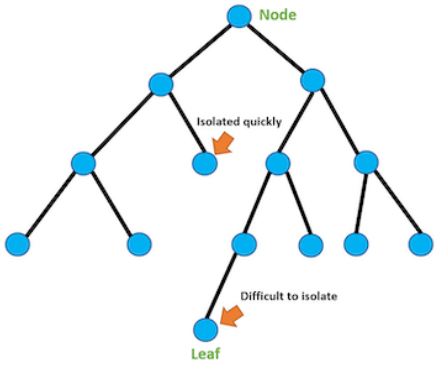

Fig. 20 An example of isolation forest process

The Isolation Forest isolate observations into abnormal and normal observations by the following steps:

1. At each node in the decision tree, randomly select a feature then selecting a splitting value between maximum and minimum values of the selected feature/column to split the dataset into two.
2. After that, keep doing the above step until all observations are isolated from each other.

The anomalies are detected by finding the observation that is isolated faster or easier than other observations because the anomalies are normally far from normal.

5.4 Apply One-hot encoding on categorical columns

- Gender and Offer type columns

5.5 Declare X (features) and y (a target)

- Features: other columns except for the target
- Target: the response column

5.6 Split the built data into training and testing data

Split the built data into training and testing data by keeping the target ratio when splitting (stratify).

	reward	difficulty	duration	email	mobile	social	web	age	income	gender_F	gender_M	gender_O	offer_type_bogo	offer_type_discount	offer_type_informational
0	2	10	7	1	1	0	1	68	70000.0	0	1	0	0	1	0
1	0	0	4	1	1	0	1	68	70000.0	0	1	0	0	0	1
2	5	5	7	1	1	0	1	68	70000.0	0	1	0	1	0	0
3	2	10	10	1	1	1	1	68	70000.0	0	1	0	0	1	0
4	5	5	5	1	1	1	1	65	53000.0	0	1	0	1	0	0

Fig. 21 An example of the training dataset after cleaning process.

Implementation

The implementation processes are as follow:

1. Load Portfolio, Profile, and Transcript datasets.
2. Clean the datasets according to the data preparation steps above.
3. Train the XGBoost classifier according to the Algorithms and Techniques part.
4. Validate accuracy by eval_metric. The validation accuracy is determined by AUC according to the Benchmark section.
5. If the accuracy doesn't improve for early_stopping_rounds value rounds, stop training the model. Otherwise, return to step 3.

Refinement

As mentioned in the Benchmark section, the XGBoost classifier without tuning hyperparameters has an accuracy around 71%.

To increase the accuracy of the model, GridSearchCV is implemented to find optimal hyperparameters for the model.

GridSearchCV

Search the optimal hyperparameters for the XGBoost classifier from the lists of hyperparameters (grid). The optimal hyperparameters are selected by checking the model performance (accuracy) with the cross-validation method.

```
param_grid = {
    'max_depth': [3, 4, 5],
    'learning_rate': [0.05, 0.1],
    'gamma': [0],
    'reg_lambda': [0],
    'min_child_weight':[0,1]
}
```

Fig. 22 An example of GridSearchCV grid

Note that the XGBoost classifier hyperparameters are already described in the Algorithms and Techniques section.

The GridSearchCV used parameters are as follow:

- estimator – the estimator which is the XGBoost classifier with binary logistic
- param_grid – lists of hyperparameters.
- scoring – performance metric (AUC)

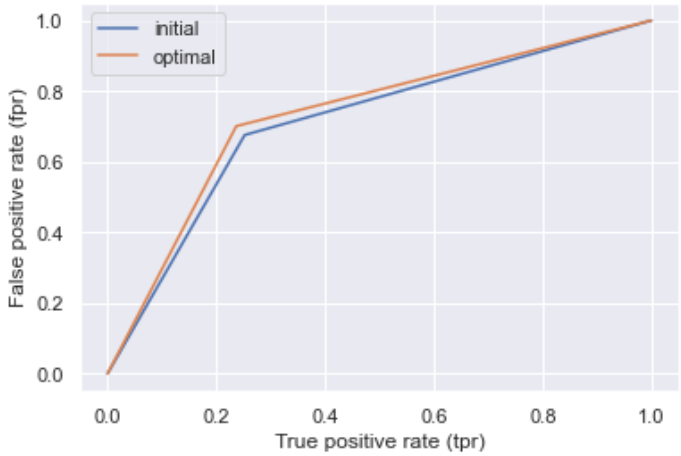

Fig. 23 The ROC graph between the initial model and optimal model

The ROC graph above shows the ROC graph between the initial model and the optimal model. It can clearly see that the optimal model is a better model in terms of AUC.

The accuracy of the optimized model is about 73% which slightly increases (2%) from the initial model.

Results

Model Evaluation and Validation

The testing dataset was used to evaluate the model. To get the best model performance, the optimized hyperparameters from GridSearchCV are applied to the model.

The final model's qualities are as follow:

- The gamma for pruning tree is 0
- The L2 regularization lambda is 0.5
- The learning rate is 0.c2
- The maximum tree depth is 11
- The model estimator is a binary logistic
- The minimum number of residuals in each leaf is none or 0
- The optimized model performs 99 iterations for the training section
- The optimized model doesn't improve at the 89th iteration

To verify the robustness of the optimal model, the tests below are applied.

1.The ratio of target

The target ratio should be reasonable to make the model robust to overfitting. Note that the skewed target values often make the overfitting model.

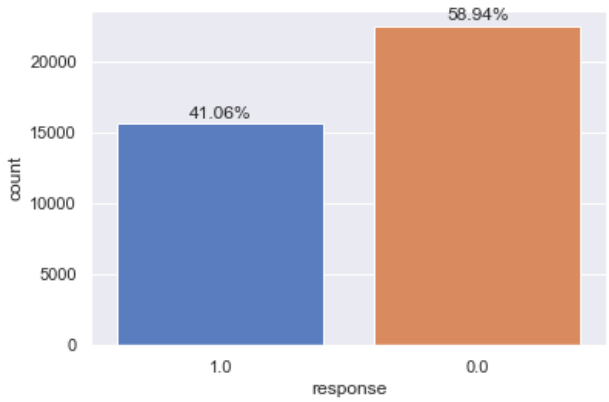

Fig. 24 The ratio of target value (response)

According to the plot, the target proportion is reasonably skew (not much skew).

2. The difference between the AUC value in the training and validation datasets

To prevent an overfitting problem, the AUC comparison of training and testing data is observed. The models that have a testing AUC close the testing AUC often indicate an overfitting problem. To measure this, the comparison of the training AUC and testing AUC value is used.

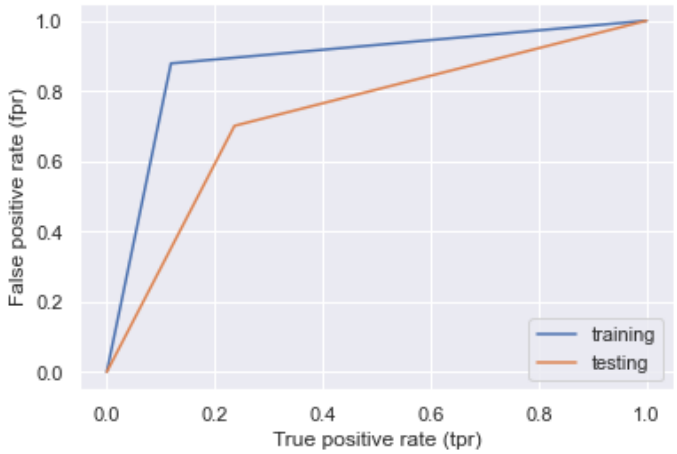

Fig 25. The comparison of the training AUC and testing AUC value by ROC graph

From the ROC graph, the training data AUC not close with testing data AUC. Therefore, the model is robust to an overfitting problem.

Justification

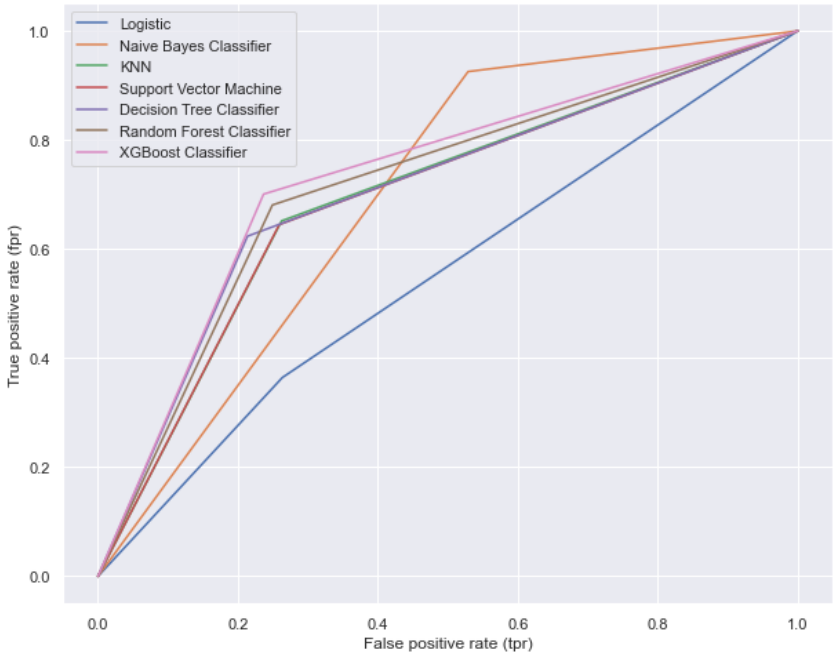

Fig. 26 The ROC graph of candidate classifier models

	Model	Accuracy
	XGBoost Classifier	73.76
	Random Forest Classifier	72.45
	KNN	70.23
	Support Vector Machine	70.14
	Naive Bayes Classifier	65.76
	Logistic	58.37
	Decision Tree Classifier	58.37

	Model	AUC score
6	XGBoost Classifier	0.73
5	Random Forest Classifier	0.72
1	Naive Bayes Classifier	0.70
4	Decision Tree Classifier	0.70
2	KNN	0.69
3	Support Vector Machine	0.69
0	Logistic	0.55

Fig. 27 The tables of Accuracy and AUC score of each candidate classifier

From the ROC graph and the tables above, the best performing model is the XGBoost Classifier in terms of accuracy and AUC values. However, it hard to conclude which models are better because the top two models have accuracy and AUC close to each other. To find more which model is significantly better in classification, the comparison of the two top models is performed by the confusion matrix below.

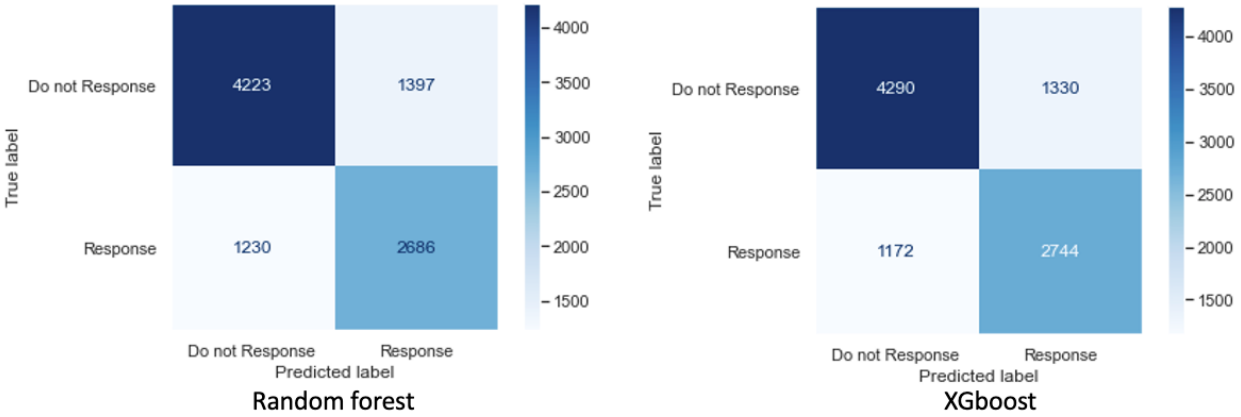

Fig. 28 The confusion matrix of Random forest and XGBoost

According to the confusion matrix, the XGBoost classifier correctly classifies the responded and not responded at 70.07% and 76.33% while Random forest is 68.59% and 75.14%. The XGBoost outperforms Random Forest in classifying responded and not responded by 1.48% and 1.19%. Therefore, the conclusion can be performed that the best performing model is XGBoost Classifier in both classifications.

Conclusion

The prediction can tell a company what offers are preferred by an individual customer. The customer demographic and offer details are put to the model to make a prediction. The input values are as follow:

customer demographic

- age (int) – customer age
- gender (string) – customer gender; either M, F or O. The 'O' value is for other genders
- income (float) – income from customer

offer details

- offer_type (string) – type of offer: either BOGO, discount or informational
- difficulty (int) – minimum dollars that require to spend to get an offer
- reward (int) – reward that customer received after completed an offer
- duration (int) – time that an offer available for completing the offer
- channels (list of strings) – list of channels that can receive an offer (web, email, mobile, social)

An example of prediction

	gender	age	income
0	F	43	40000

Fig. 29 The customer demographic table

The table shows the customer demographic who will be predicted by the Starbucks offers. After predicting, the results are as follow:

The customer responds to these offers

	reward	channels	difficulty	duration	offer_type	id	offer_id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	4
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2	6
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4	7
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5	10

Fig. 30 The offers that are responded by the customer

The customer does not respond to these offers

	reward	channels	difficulty	duration	offer_type	id	offer_id
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	2
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed	3
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7	5
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837	8
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d	9

Fig. 31 The offers that are not responded by the customer

Reflection

These are the problems or challenges for me to doing this project:

The problem that I find the most difficult is in building the training dataset section. Finding the criteria of what offers are responded or not responded by the individual customers is hard because you need to find what offers are actually responded. For example, the customer might receive an offer and accidentally complete an offer without view the offer.

And also, in the cleaning section, when creating one-hot encoding for channels column. The get_dummies method can't be performed because it has an invalid type of input values which is a list inside the series. Therefore, I need to find a way to convert that categorical values to numeric or one-hot encoding manually.

The problem also occurs in the outlier removal part. The common method which is the z-score is not enough to remove an outlier. Therefore, the other method needs to implement to remove outlier, which I end up with isolation forest.

The abnormal missing values are also the problem. I need to investigate according to the data exploration and visualization section.

The reasons above can conclude that the main problems or challenges for me are in the data preprocessing and cleaning sections, even though the modeling section also has a problem. For instance, I need to find a way to compare which models give an excellent accuracy for classification.

Improvement

Considering the values that are dropped along with missing values, those values might be useful for the model. The model might be less accurate than it actually was when that values are not included in the model training process.

To solve this problem, the imputation method can be performed. This method brings back the values that are dropped along with the missing values by imputing the missing values. Although the imputation method might improve the model accuracy, beware that the shoddy imputation makes the model less accurate.

The difference of GridSearchCV cross-validation fold number, the split ratio of training and testing dataset and evaluate metric, might increase the model accuracy.