Machine Learning Engineer Nanodegree

Sharmin Ahmed

October 26, 2020

Starbucks Capstone Project Report

Project Overview

This is a Starbucks capstone project from Udacity Machine Learning Engineer Nanodegree program. The project data contains a simulated data set that mimics customer behavior on the Starbucks rewards mobile app. Unlike the actual data, this data set only includes customer behavior concerning one product. For company profit and building a better user experience, Starbucks has to send offers the corresponding user is more likely to complete.

Problem Statement

The problem is to determine what offers to send to which customers. Determination can be based on their history, such as previous purchases, genders, age groups, locations, etc. The project's goal is to predict if a user will complete an offer within the given timeframe.

Datasets

The dataset for this project is provided by Udacity and Starbucks. This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app, which is included in three files: portfolio.json, profile.json and transcript.json. These data sets have to be cleaned and merged into one data frame for model purposes.

Evaluation Metrics

Since this is a classification problem, the Following evaluation matrices are used:

- **Precision:** It is the number of true positives divided by the number of true positives, plus the number of false positives.
- **Recall:** It is the number of true positives divided by the number of true positives plus the number of false negatives.

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

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- **F-1 score:** F-1 score is a way of combining the precision, and it is defined as the harmonic mean of the model's precision and recall.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Analysis

I. Data Exploration and Cleaning

Upon exploration of profile, portfolio and transcript dataset, there are several insights.

portfolio:

	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

- There are three types of offers : 'bogo', 'informational' and 'discount'. These data would be better for modeling, if applied one hot encoding method.
- ‘channels’ column needs to be processed. Information is be extracted and also applied with one hot encoding method.
- ‘duration' column should be converted into hours as 'time' in transcript.json is also in hours.
- id needs to be renamed to offer_id to avoid confusion.

after cleansing portfolio:

	difficulty	duration_hour	offer_id	reward	web	email	mobile	social	bogo_offer	informational_offer	discount_offer
0	10	168	ae264e3637204a6fb9bb56bc8210ddfd	10	0	1	1	1	1	0	0
1	10	120	4d5c57ea9a6940dd891ad53e9dbe8da0	10	1	1	1	1	1	0	0
2	0	96	3f207df678b143eea3cee63160fa8bed	0	1	1	1	0	0	1	0
3	5	168	9b98b8c7a33c4b65b9aebfe6a799e6d9	5	1	1	1	0	1	0	0
4	20	240	0b1e1539f2cc45b7b9fa7c272da2e1d7	5	1	1	0	0	0	0	1

profile:

	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

- The dataset has 2175 missing values on: 'gender', 'income' column. Corresponding age column of those records are set to default 118. These missing values are cleaned.
- There are three 'gender' categories- M (8484), F (6129) and O (212). O could be associated with missing value records.
- Income is grouped into 'average', 'above_average', 'high' buckets and 'age' is grouped into 'young_adult', 'middle_aged', 'old', 'elderly' buckets for model purpose.
- membership_days and member_category: 'regular', 'long_term' are calculated from 'became_member_on column'.
- Id is renamed to customer_id.

after cleansing profile:

	age	became_member_on	gender	customer_id	income	membership_days	age_group	income_range	member_category
1	55.0	2017-07-15	F	0610b486422d4921ae7d2bf64640c50b	112000.0	1198	old	high	regular
3	75.0	2017-05-09	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0	1265	elderly	high	regular
5	68.0	2018-04-26	M	e2127556f4f64592b11af22de27a7932	70000.0	913	elderly	above_average	regular
8	65.0	2018-02-09	M	389bc3fa690240e798340f5a15918d5c	53000.0	989	elderly	above_average	regular
12	58.0	2017-11-11	M	2eeac8d8feae4a8cad5a6af0499a211d	51000.0	1079	old	above_average	regular

transcript:

	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'}

- 'person' is renamed to 'customer_id'.

- The dataset has no missing values.
- The 'value' column is a dictionary and needs to be proccessed. offer_id and amount is extracted to newly created corresponding columns.
- All events in this dataset are: 'transaction', 'offer received', 'offer viewed' and 'offer completed'. Offers are processed with one hot encoding and transaction event are extracted to transaction_df
- other events excluding transaction are extracted to offer_df.

after cleansing transcript:

transaction_df:

	customer_id	time	amount
12654	02c083884c7d45b39cc68e1314fec56c	0	0.83
12657	9fa9ae8f57894cc9a3b8a9bbe0fc1b2f	0	34.56
12659	54890f68699049c2a04d415abc25e717	0	13.23
12670	b2f1cd155b864803ad8334cdf13c4bd2	0	19.51
12671	fe97aa22dd3e48c8b143116a8403dd52	0	18.97

offer_df:

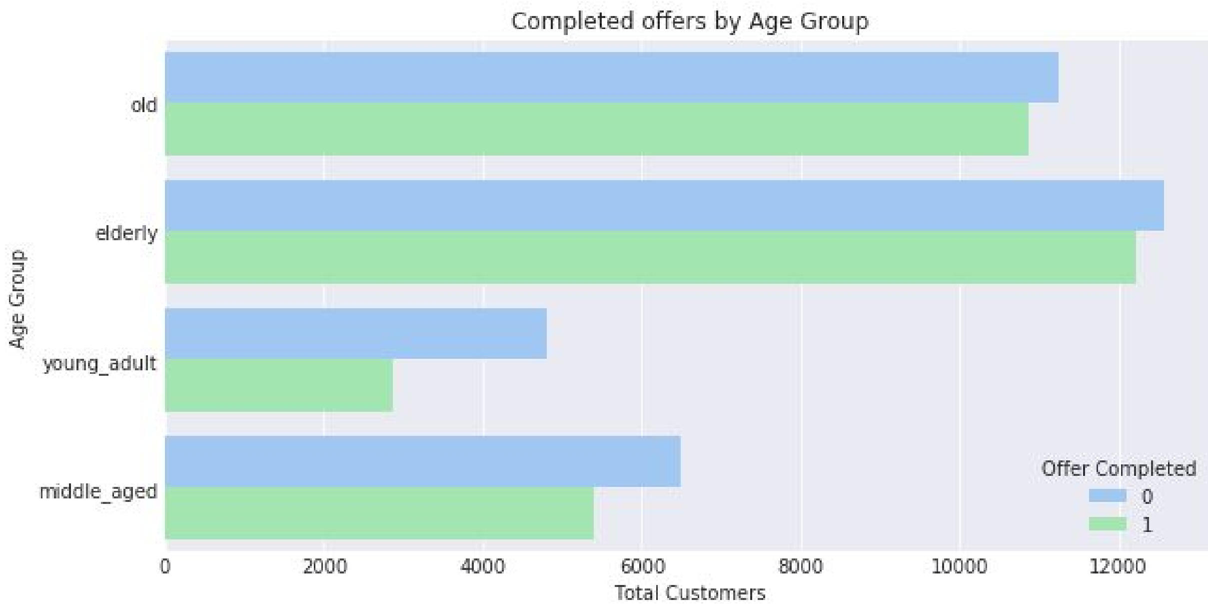
	customer_id	offer_id	time	offer completed	offer received	offer viewed
306497	a6f84f4e976f44508c358cc9aba6d2b3	2298d6c36e964ae4a3e7e9706d1fb8c2	714	1	0	0
306506	b895c57e8cd047a8872ce02aa54759d6	fafdcd668e3743c1bb461111dcafc2a4	714	1	0	0
306507	8dda575c2a1d44b9ac8e8b07b93d1f8e	0b1e1539f2cc45b7b9fa7c272da2e1d7	714	0	0	1
306509	8431c16f8e1d440880db371a68f82dd0	fafdcd668e3743c1bb461111dcafc2a4	714	1	0	0
306527	24f56b5e1849462093931b164eb803b5	fafdcd668e3743c1bb461111dcafc2a4	714	1	0	0

merged_data:

Once the cleaning is done, the datasets are merged into a big data frame in order to analyze the relation between customer behaviour on a offer. 'money_spent' and 'offer_completed' attributes are also calculated on the process.

II. Exploratory Visualization

- **Offer completion by age_group:**

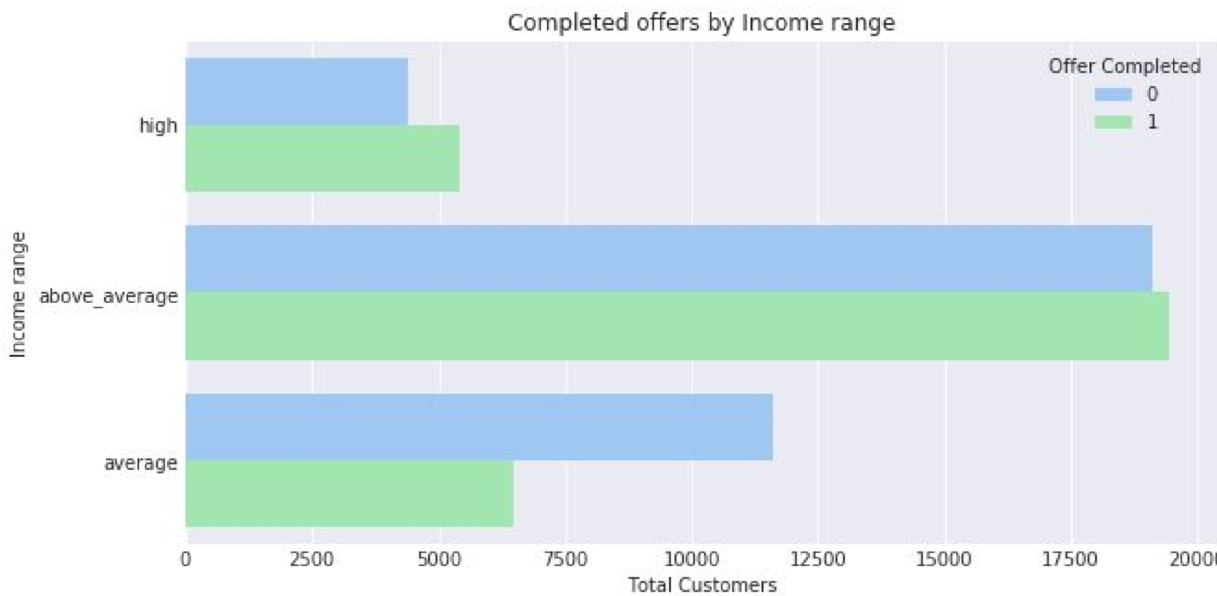


age_group are categrozed into - young_adult (17-30) , middle_aged (31-45), old (46-60) and elderly (60-105). From the visualization we can see 45- 105 aged people completed maximum offers.

- **Offer completion by Gender:**

Male numbers are more in the dataset. Hence, Male completed more offers also. Although not completed offer numbers were also higher in Males.

- **Offer completion by Income range:**



Income range are categorized into average (29999-50000), above_average (50001-90000), and high (90001-120001) category. People with above_average income completes more offer.

- **Offer completion by Membership year:**

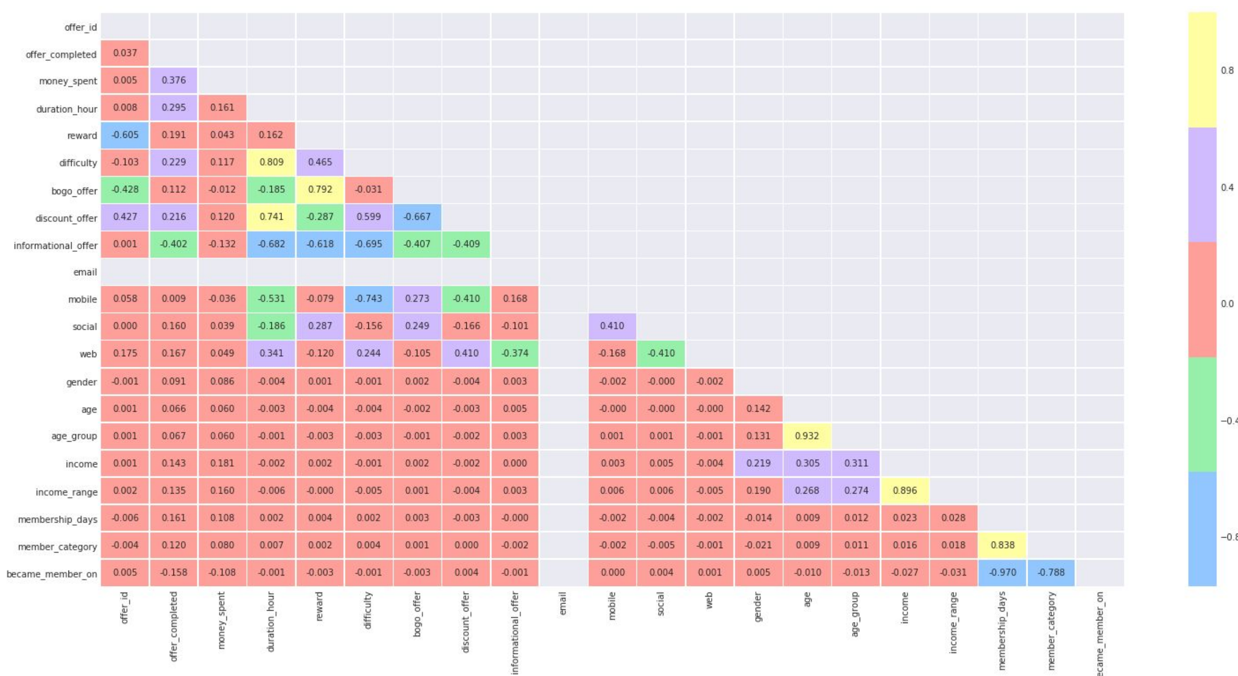
Most members are from 2016-2018. 2016-2017 members have higher offer completion rate.

III. Feature Engineeing

- **Encoding categorical columns with numerical values:** For model purpose, the categorical valued columns are encoded with numerical values. The prospective columns are : gender, age_group, income_range, member_category, offer_id.

- **Normalizing features:** The numerical valued columns 'money_spent', 'duration_hour', 'reward', difficulty', 'income', 'membership_days' are normalized for better fitting.

- **Correlation matrix:** A heatmap of the correlation matrix of merged_data is plotted to select best features.



- ❑ After visualization, 'money_spent', 'duration_hour', 'reward', 'difficulty', 'bogo_offer', 'discount_offer', 'informational_offer', 'social', 'web', 'member_category', 'became_member_on', 'gender', 'age_group', 'income_range' attributes are selected as independent variables.
- ❑ For dependent variable offer_completed attribute is selected.

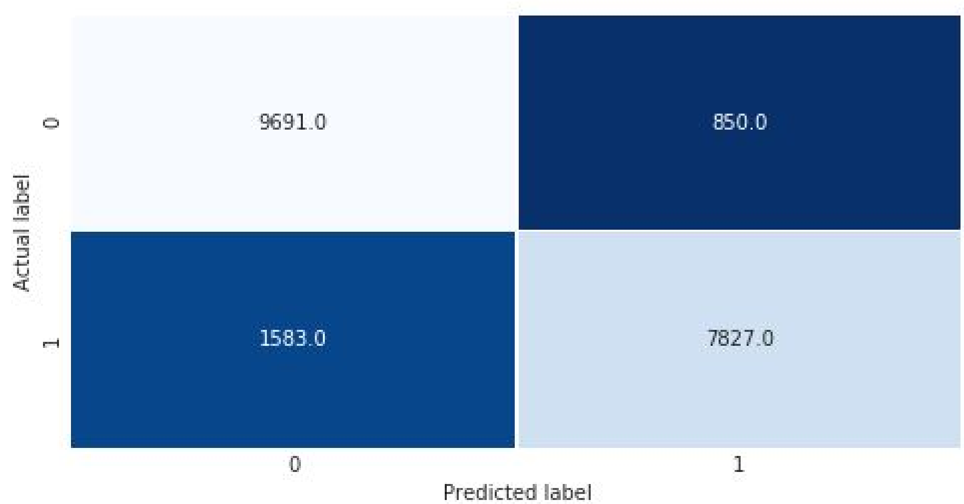
IIII. Model Selection and Tuning

Four models are selected for the problem : Logistic Regression, Random Forest Classifier, Support Vector Classification (SVC) and Gradient Boosting Classifier. From which Logistic Regression is selected as the benchmark model.

The models were tuned and best fitted models were tested with test_set.

Logistic Regression:

- Tuned parameters:
  I. 'penalty': ['l1','l2'] → l1
  II. 'C': [0.001,0.01,0.1,1,10,100,1000] → 100
- Confusion matrix:



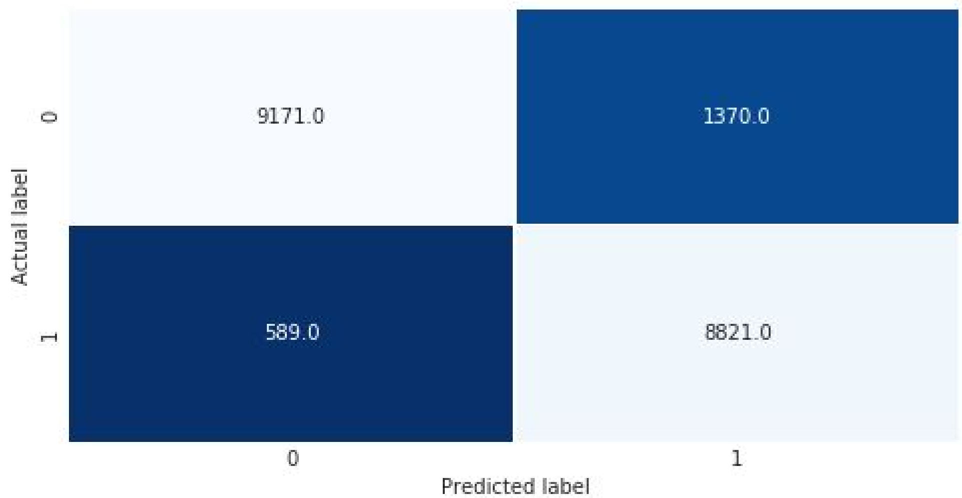
- Classification report:

	precision	recall	f1-score	support
0	0.86	0.92	0.89	10541
1	0.90	0.83	0.87	9410
avg / total	0.88	0.88	0.88	19951

Random Forest Classifier:

- Tuned parameters:
  I. 'max_depth': [3, 5] → 5
  II. 'min_samples_leaf': [2, 3] → 2
  III. 'n_estimators': [100, 1000, 2000] → 100

- Confusion matrix:



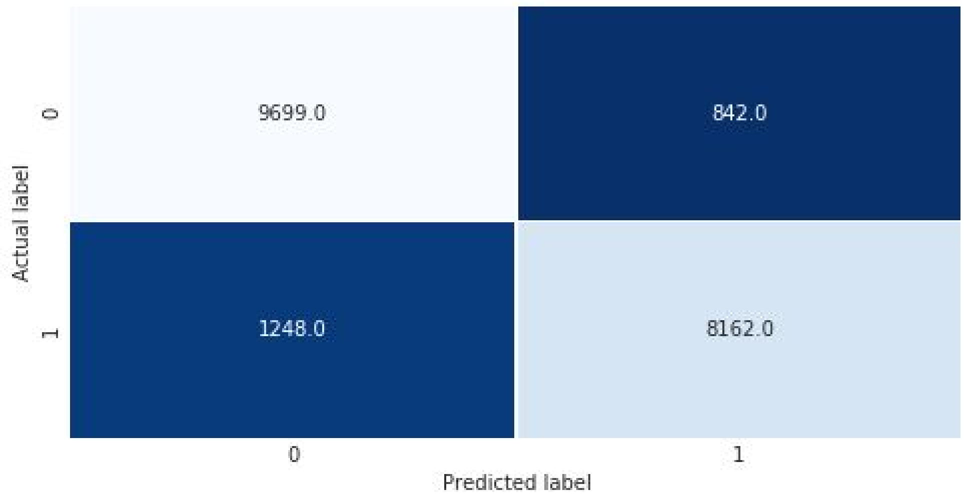
- Classification report:

	precision	recall	f1-score	support
0	0.94	0.87	0.90	10541
1	0.87	0.94	0.90	9410
avg / total	0.90	0.90	0.90	19951

Support Vector Classification (SVC):

- Tuned parameters:
    I. 'C': [10, 100, 1000] → 1000
    II. 'gamma': [1]

- Confusion matrix:



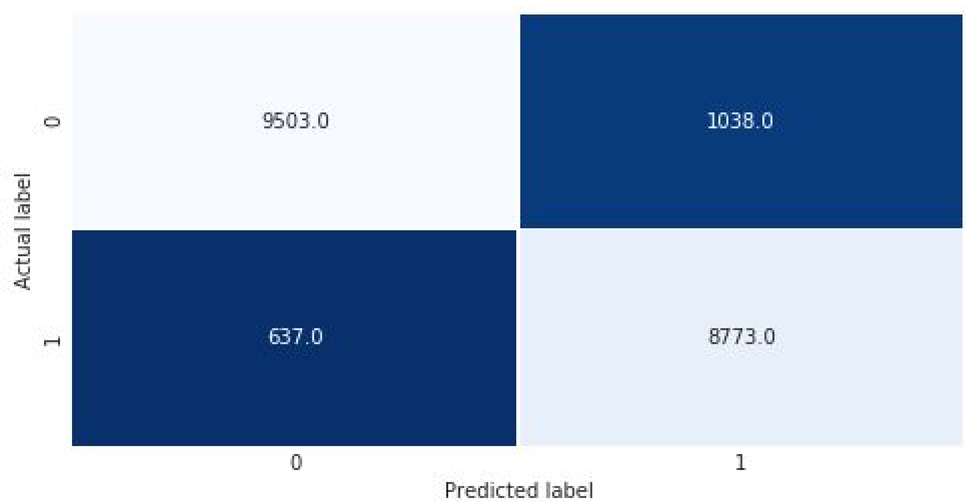
- Classification report:

	precision	recall	f1-score	support
0	0.89	0.92	0.90	10541
1	0.91	0.87	0.89	9410
avg / total	0.90	0.90	0.90	19951

Gradient Boosting Classifier:

- Tuned parameters:
  I. "learning_rate": [0.1]
  I. "min_samples_split": [2]
  II. "min_samples_leaf": [3]
  III. "max_depth": [3, 5] → 5
  IV. "subsample": [0.95, 0.1] → 0.95
  V. "n_estimators":[100, 1000] → 100

- Confusion matrix:



- Classification report:

	precision	recall	f1-score	support
0	0.94	0.90	0.92	10541
1	0.89	0.93	0.91	9410
avg / total	0.92	0.92	0.92	19951

IV. Model Comparison

The benchmark model's precision, recall, f1-score were 0.88. RandomForest and SVM performed a little better achieving 0.90. The best performance was given by Gradient Boosting Classifier 0.92.

Model	Precision	Recall	f1-score
Logistic Regression (Benchmark)	0.88	0.88	0.88
Random Forest Classifier	0.90	0.90	0.90
SVM Support Vector Classification	0.90	0.90	0.90
Gradient Boosting Classifier	0.92	0.92	0.92

From the result, it's clear that selected feature performed well in predicting offer_completed.

V. Conclusion

Based on the experiment conducted in this project, the best classifier is Gradient Boosting Classifier. Though there are some limitation as only several parameters were tuned for computation power issues. So, for further project for refining the model, I suggest to do further feature engineering and better hyperparameter tuning so that other models, such as, SVC, Logistic Regression, Random Forest Classifier can comprehend this dataset better.