CAPSTONE PROJECT

Starbucks Customer Segmentation Project

JANUARY 23, 2022

AUTHOR: DHANALAKSHMI C.
For: AWS MLE NANODEGREE

Table of Contents

Definition

For the capstone, I chose to work on Starbucks dataset provided by Udacity as I found it interesting and there are scopes for analysing huge data which can be used for addressing many demanding problems in the real-world.

Project Overview

The project aims on predicting how customer responds to an offer in Starbucks app. This is one of the marketing strategies that every company may follow to get hold of their existing customers. 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.

There are 3 types of offers (BOGO, Discount, Informational) where each offer type in turn contains 3-4 offers. All offers have different expiry time / reward / difficulty to validate the user's responses to various offers. As an example, a BOGO offer might be valid for only 5 days. The duration feature in the data set explains that informational offers have a validity period even though these ads are merely providing information about a product; for example, if an informational offer has 7 days of validity, it can be assumed that the customer is feeling the influence of the offer for 7 days after receiving the advertisement.

To give an example, a user could receive a discount offer buy 10 dollars get 2 off on Monday. The offer is valid for 10 days from receipt. If the customer accumulates at least 10 dollars in purchases during the validity period, the customer completes the offer.

There is a to note in this dataset: Customers do not opt into the offers that they receive; in other words, a user can receive an offer, never actually view the offer, and still complete the offer. For example, a user might receive the "buy 10 dollars get 2 dollars off offer", but the user never opens the offer during the 10-day validity period. The customer spends 15 dollars during those ten days. There will be an offer completion record in the data set; however, the customer was not influenced by the offer because the customer never viewed the offer.

Each offer is weighed with reward points (0, 2, 3, 5, 10) that can be claimed to avail more benefits from Starbucks. More details on rewards are available here.

Problem Statement

There are also customers who have used the app for making a purchase but never received or seen the offers. If the company has decided to send an offer to these customers, it is expected to address questions such as 'what offer can be sent? To which demographic groups the offer can be sent? If these customers will complete the offers once received?'. Therefore, the problem statement is defined here to find if the offer sent is **successful or not**. This helps the business to target potential customers – i.e., **customer targeting**.

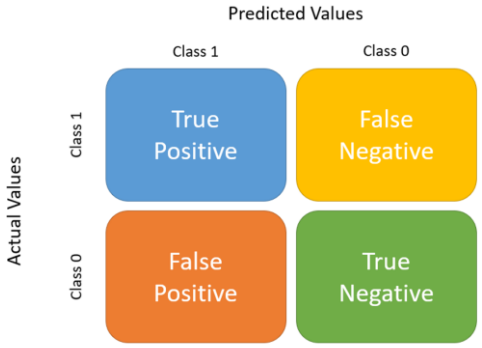
What is customer targeting? At its most basic, customer targeting is the act of reaching out to a portion of your customer list to re-engage them and drive sales. An interesting study about the evolution of customer targeting is explained in this article.

It is also expected to use the **AWS ML workflow** concepts learnt in Udacity AWS MLE nanodegree program, so far. A similar approach to following the AWS ML workflow in Sagemaker Studio can be found here.

Metrics

The classification problems are usually evaluated with a **confusion matrix** which helps in understanding the prediction's accuracy, precision, recall, F1 score etc. In this project, the target is expected to have two classes (offer successful, offer not successful). Hence, the confusion matrix will have a dimension of 2*2.

The benchmark metrics for evaluating the model performance is **Confusion Matrix** which in turn helps in finding accuracy, precision, recall, F1 score etc. The Benchmark accuracy score is set as **90%** for this project.



Analysis

Data Exploration

The datasets were used from Udacity classroom as below:

- portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
- profile.json - demographic data for each customer
- transcript.json - records for transactions, offers received, offers viewed, and offers completed

All three datasets did not have any missing values except for a few outliers in profile dataset. Each dataset was fed into a pandas dataframe and their dimensions were identified as:

- Portfolio dimension: (10, 6) – 10 records * 6 features
- Profile dimension: (17000, 5) – 17000 records * 5 features
- Transcript dimension: (306534, 4) – 306534 records * 4 features

Here is the schema and explanation of each variable in the files:

portfolio.json

- id (string) - offer id
- offer_type (string) - type of offer i.e., 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)

profile.json

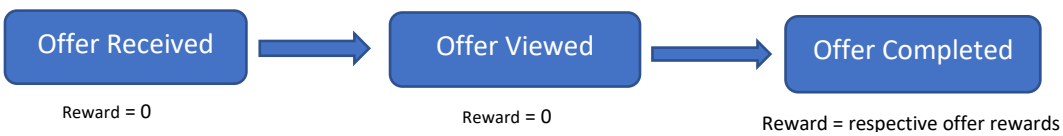
- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

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

Exploratory Visualization

Each dataset was separately cleaned and analysed. There were then carefully merged and explored again. The transcript dataset had an event column with values ‘offer received’, ‘offer viewed’, ‘offer completed’, ‘transaction’. The records with transaction were carefully reserved, cleaned, assigned offers to be fed as test dataset for later use.

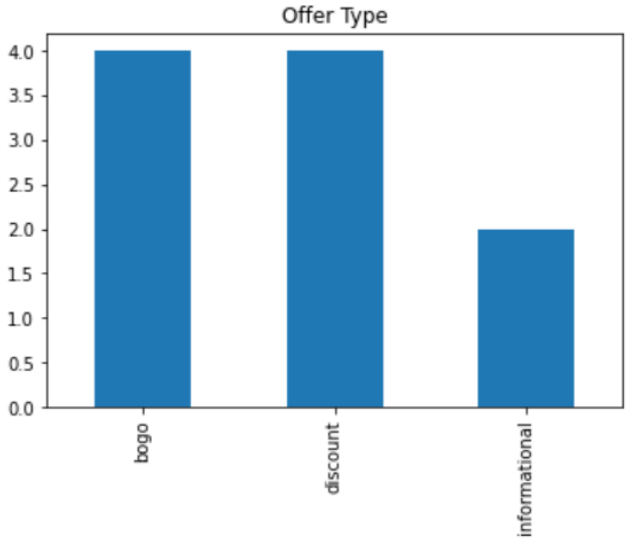


Portfolio

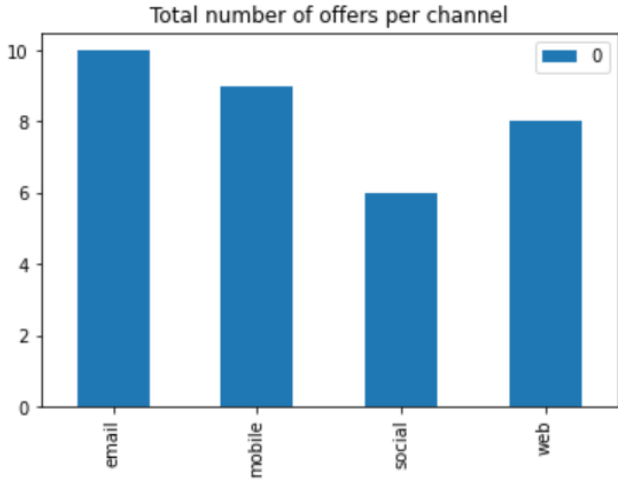
The id column in portfolio dataset represented offer id and hence renamed the column as ‘offer_id’ in the later analysis. Also, they were mapped to real numbers to easily categorize.

	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
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

Most of the offers are grouped in BOGO and Discount offer types. There are couple of offers under informational category. However, they don't have any reward or difficulty points.

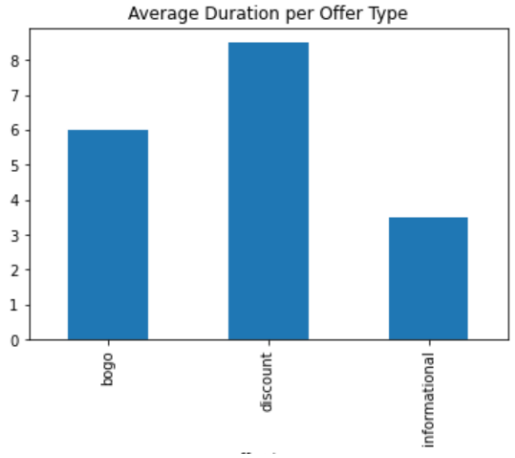


All the offers were sent by email. However, there were other channels such as mobile, social media, web through which the offers were sent to the customers. The least used channel was social media.



There is no correlation between reward and difficulty. They were set for a business purpose in each offer.

The Discount offer types stayed for a longer time than other offer types on an average.



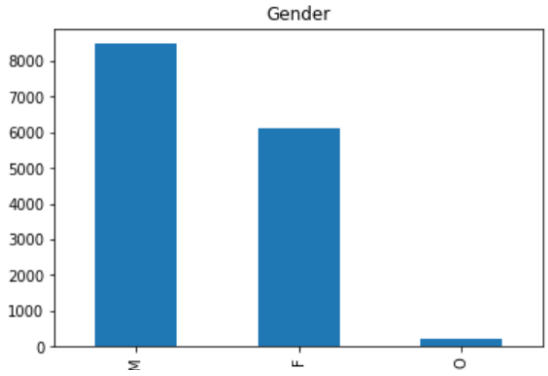
Profile

The profile dataset had null values in gender and income whose age is 118. This age stood as an outlier for a demographic analysis. Hence, it adds no value to the dataset even if the null values are handled. Hence those records were safely removed from the dataset.

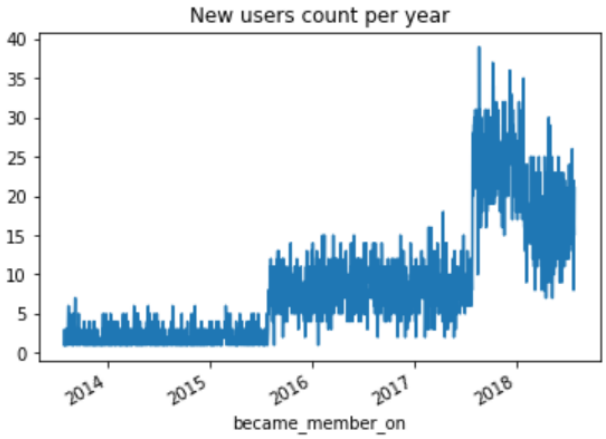
The id column in profile dataset represented members of the app and so, it was renamed as member_id.

	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

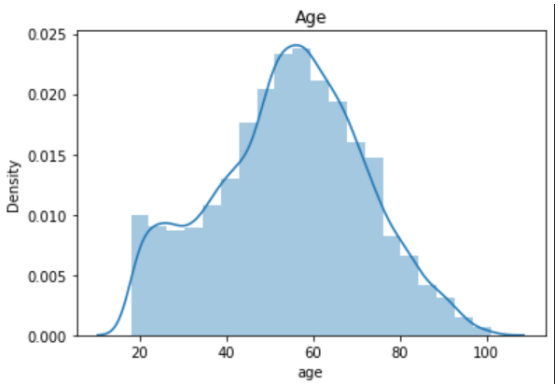
Considering the gender category in the dataset, there seemed to be more men than other genders who responded to the offers and also did transactions. Around 212 people selected others category.



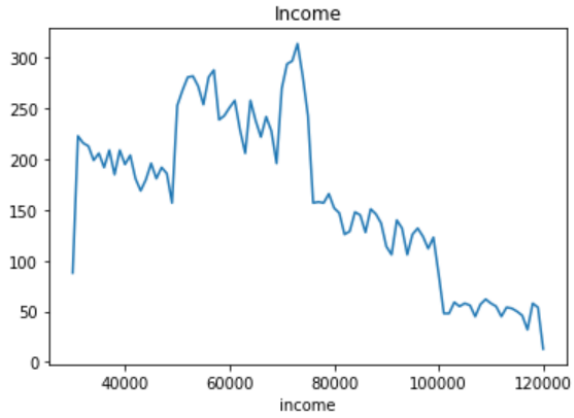
Based on became_member_on column, it was identified that every couple of years, the new addition of membership doubled. There is a huge rise of membership between end of 2017 and the start of 2018 and then reduced again.



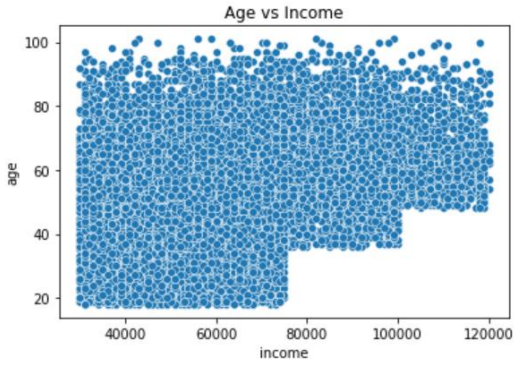
There was almost a normal distribution in the age category. Those with age between 50-60 responded more for offers and did transactions than other age groups.



The average income of a member in the dataset was approximately 65000.



There was no relationship between age and income. Some people at early 20s were also earning same as other age groups.

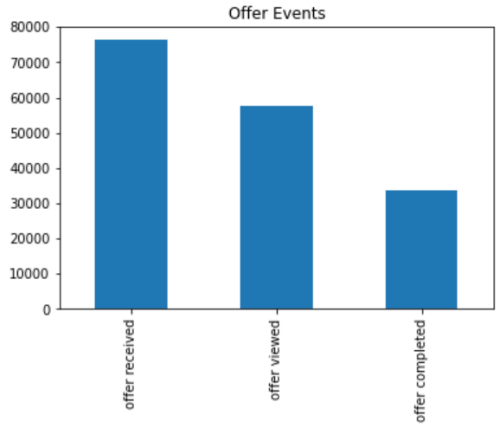


Transcript

The value column in transcript dataset had a dictionary of keys 'offer id', 'offer_id', 'reward', 'amount' that were inconsistently available throughout the dataset. They were then exploded into different columns.

	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

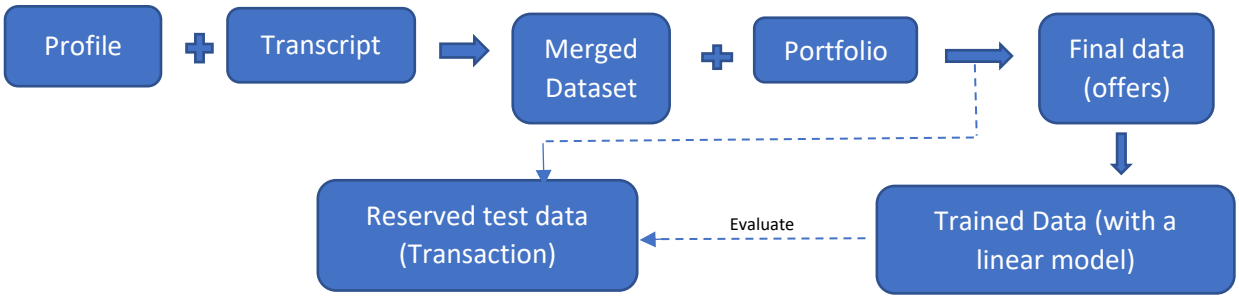
Only 50% of the received offer were completed. The event feature was renamed as **offer_successful** and was assigned a value of 0 for 'offer received' and 'offer viewed'. Those with 'offer completed' event was given a value of 1. The offer_successful feature was then set as the **target** column for the project.



There were 10 offers sent to the customers. Most of the customers responded to offer number 6 and 7 that are under discounts. The least responded offer is 3 which is of informational offer type.

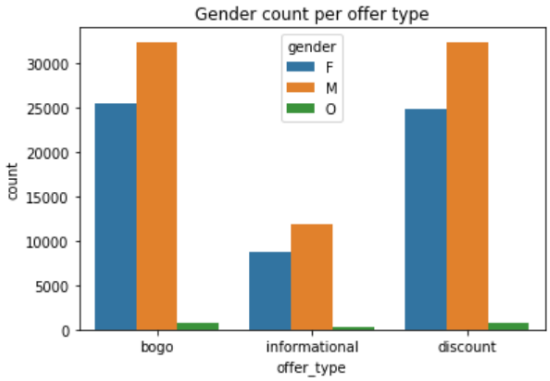
Merged Datasets

The profile and transcript datasets were initially merged and analysed. The merged dataset had null values. They were carefully handled and pre-processed. They were then merged with portfolio dataset and the final dataset was feature engineered to feed into the model.



Profile-transcript

After merging the profile and transcript datasets, it was identified that almost equal set of users responded to both bogo and discount offer types.



Algorithms and Techniques

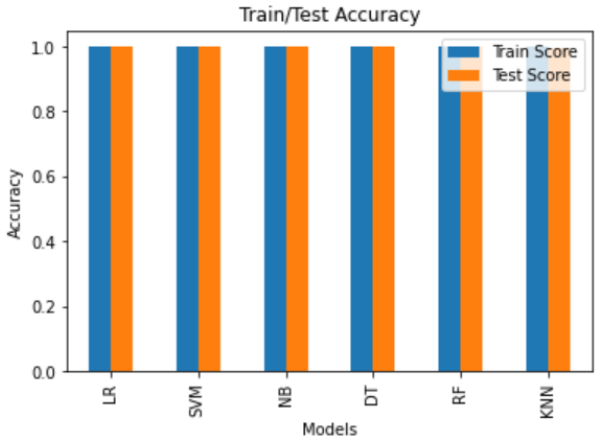
The merged dataset was then split into train and test dataset to be evaluated with various machine learning algorithms. There were algorithms which required scaled data and so the train and test datasets were scaled separately.

Below were the machine learning classifier algorithms used for the initial evaluation.

- Logistic Regression
- K-Nearest Neighbors
- Support Vector Machine
- Naive Bayes
- Decision Tree
- Random Forest

Benchmark

Each algorithm was trained with the train dataset and then evaluated with a test dataset (which is technically a validation dataset). Except KNN, all other algorithms yielded an accuracy score of 100%. This may lead to over-fitting. Hence, the dataset was trained with a linear learner model in Sagemaker and then the pre-processed transaction records (that the model never seen data) was later fed into the model for evaluation.



	Logistic Regression	Support Vector Machine	Naïve Bayes	Decision Tree	Random Forest	K-Nearest Neighbours
Confusion Matrix	[[21066 0] [0 6470]]	[[21066 0] [0 6470]]	[[21066 0] [0 6470]]	[[21066 0] [0 6470]]	[[21066 0] [0 6470]]	[[21063 3] [113 6357]]

Methodology

Data Pre-processing

Below are the tables representing the type of pre-processing done in each dataset

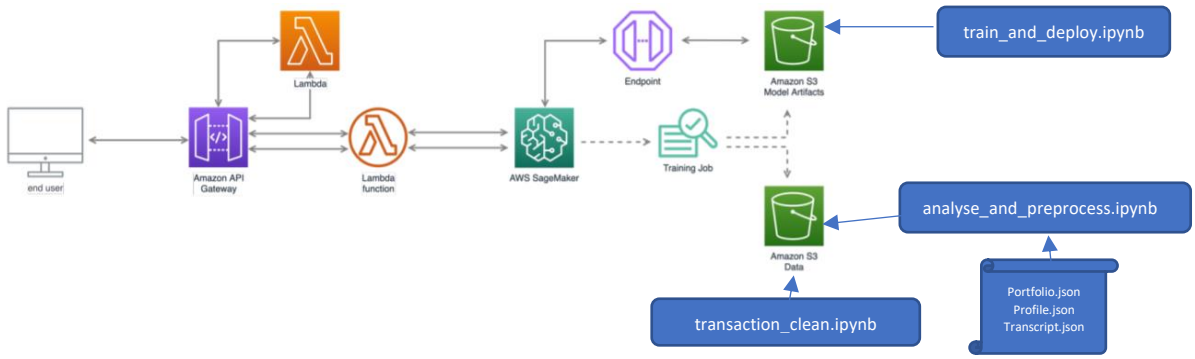
Dataset	Column Name	Null Values?	Initial category	Transformed category	Final datatype
Profile	gender	Yes	M, F, O, U	0, 1, 2	Int64
	age	No	Integer	No Change	Int64
	id	No	Alphabets and numbers, dtype=Object	Renamed as member_id. Each id was mapped to an integer	Int64
	became_member_on	No	Integer	Datetime – for analysing. Calculated and used membership_days column instead of this.	Int64
	income	Yes	Float	No change	Float64
Transcript	person	No	Alphabets and numbers, dtype=Object	Renamed as member_id. Each id was mapped to an integer	Int64
	event	No	'offer received', 'offer viewed', 'transaction', 'offer completed'	'offer received': 0, 'offer viewed': 1, 'transaction': <reserved for later>, 'offer completed': 2	Int64
	value	No	Dictionary of keys ‘offer_id’, ‘reward’, ‘amount’. Yielded null values when expanded.	‘offer_id’- Strings transformed to integers ranging from 1 to 10 as per the offer ‘reward’ – removed null values ‘amount’ – not used for final dataset	Int64
	time	No	Integer	No change	Int64
Portfolio	reward	No	Integer	No change	Int64
	channels	No	List: email, mobile, social, web	Separated the list as columns having 0 and 1 as values in appropriate records	Int64
	difficulty	No	Integer	No change	Int64
	duration	No	Integer	No change	Int64
	offer_type	No	Bogo, discount, informational	0, 1, 2	Int64
	id	No	Alphabets and numbers, dtype=Object	Renamed as offer_id. Each id was mapped to an integer	Int64

Merged dataset

Dataset	Column Name	Datatype
df	difficulty	Int64
	duration	Int64
	offer_type	Int64
	offer_id	Int64
	mobile	Int32
	social	Int32
	web	Int64
	gender	Int64
	age	Int64
	member_id	Int64
	income	float64
	membership_days	Int64
	time	Int64
	reward	Int32
	offer_successful	Int64

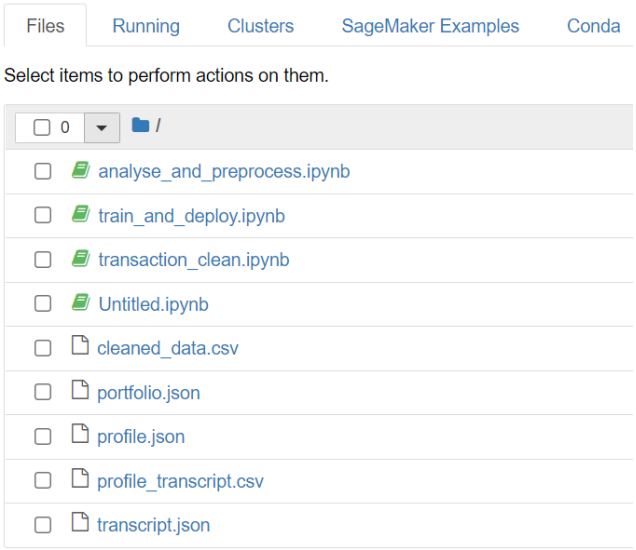
Implementation

Below is the workflow followed for this project:



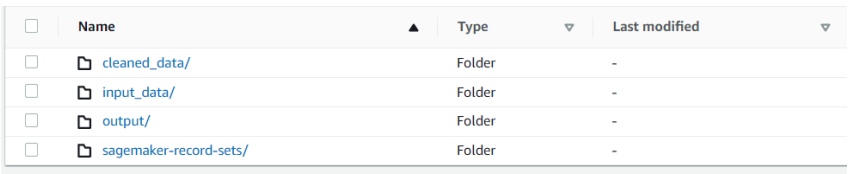
The implementation was done in three files:

- analyse_and_preprocess.ipynb
- train_and_deploy.ipynb
- transaction_clean.ipynb



- analyse_and_preprocess.ipynb

Here where the datasets are explored, cleaned and concatenated into a single dataset.
The merged dataset that was derived from profile and whole transcript dataset was reserved as profile_transcript.csv and then used by transaction_clean.ipynb
The cleaned data was uploaded to s3 bucket which was then access by train_and_deploy.ipynb



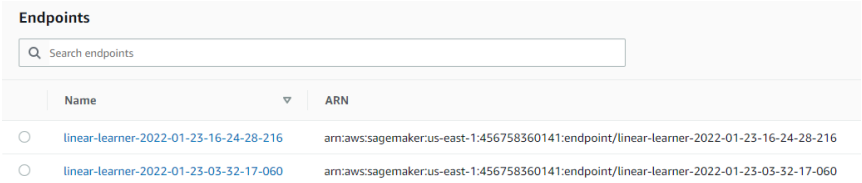
- train_and_deploy.ipynb

Here the cleaned data was scaled with which the linear learner model was trained
The parameters used for linear learner is as follow:

Parameters	Value
role	sagemaker.get_execution_role()
train_instance_count	1
train_instance_type	ml.c4.xlarge
predictor_type	binary_classifier
num_classes	2
output_path	sagemaker_session.default_bucket().output_folder
sagemaker_session	sagemaker_session
epochs	5

The model was deployed as an endpoint which was later used by lambda function

.



Results

Model Evaluation and Validation

The trained model was initially with accuracy_score and confusion_matrix which yielded a 100% score even in the dataset that was never used before (without offers – transactions – pre-processed). Hence, a different approach can also be tried to conclude if this is the best approach.

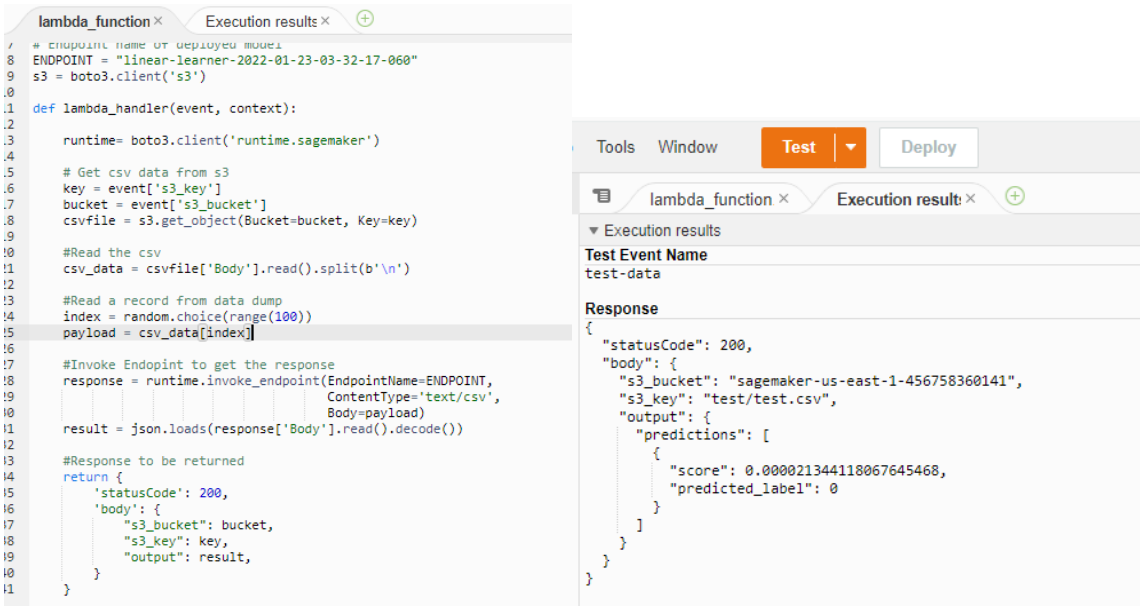
```
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy_score(y_test, lr_pred)
```

```
1.0
```

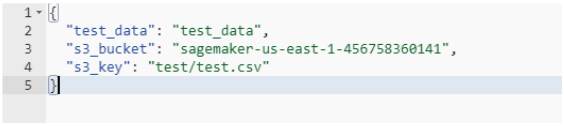
```
confusion_matrix(y_test, lr_pred)
```

```
array([[21066,     0],
       [    0,  6470]])
```

The deployed endpoint was invoked using a lambda function which used the uploaded test.csv file



Test Event:



Justification

The project is carried out with a simple and straight forward **AWS ML** workflow in a linear model to be understood by anyone. In a positive way, the datasets did not have any anomalous data and so the feature engineering was carried out in a proper way. It was built in such a way that no data was left behind. The data that did not have any offers were also utilized finally to check if the offer is successful or not. This approach is cost effective and simple.

As a workaround, the same project can be approached in a different way with deep learning concepts or by taking more features into account and predict the success rate.

Reflection

It was a pleasure working in a Udacity scholarship project with great mentors to help. Started this course with absolutely no knowledge about AWS. But then learnt too many concepts in a short span and worked on real time projects. My sincere thanks to whole Udacity and the team who helped on this scholarship.

References

https://www.starbucks.com/rewards

https://www.yourarticlelibrary.com/marketing/evolution-of-market-segmentation-approaches-explained/22185

https://towardsdatascience.com/mlops-with-mlflow-and-amazon-sagemaker-pipelines-33e13d43f238

https://en.wikipedia.org/wiki/Category:Classification_algorithms

https://www.analyticsvidhya.com/blog/2021/06/how-to-solve-customer-segmentation-problem-with-machine-learning/

https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html

https://sagemaker-examples.readthedocs.io/en/latest/introduction_to_amazon_algorithms/linear_learner_mnist/linear_learner_mnist.html

Best of all: https://www.udacity.com/course/aws-machine-learning-engineer-nanodegree--nd189