MACHINE LEARNING ENGINEER NANODEGREE

CAPSTONE PROJECT – STARBUCKS APP DATA

SHANMUKHA PRIYA MUDIGONDA

Mar 13th,2021

I. Definition

(approx. 1-2 pages)

Project Overview

Starbucks is a passionate purveyor of coffee and other beverages, headquartered in Seattle, Washington. The corporation is ranked 121st in the list of 2019 Fortune 500 companies.

They have a mobile application where registered users can use it to order coffee for pickup while mobile, pay in-store directly using the app, and collect rewards points. This app also offers promotions for bonus points to these users. The promotional offer can be merely anadvertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). This project is focused on tailoring the promotional offers for customers based on their responses.

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. 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 adiscount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks.

Not all users receive the same offer, and that was the challenge to solve with this data set.

Our task is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type. This data set is a simplified versionof the real Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products.

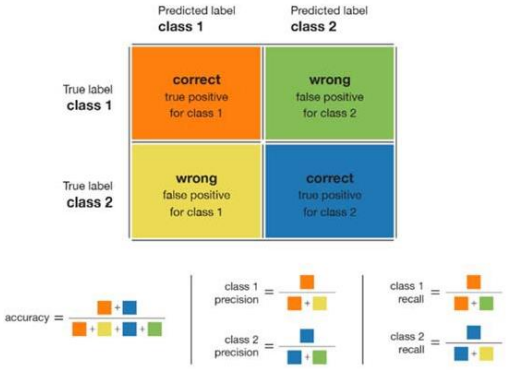
Problem Statement

The goal is that I should achieve here is to best determine which kind of offer to send to each user based on their response to the previously sent offers. Not all users receive the same offer, and that is the

challenge to solve using the data set that is provided by Starbucks, which was captured over 30 days. I will also build a machine learning model that will predict the response of a customer to an offer.

Metrics

I will consider the F1 score as the model metric to assess the quality of the approach and determine which model gives the best results. It can be interpreted as the weighted average of the precision and recall. The traditional or balanced F-score (F1 score) is the harmonic mean of precision and recall, where an F1 score reaches its best value at 1 and worst at 0



The confusion matrix and the metrics that can be derived from it.

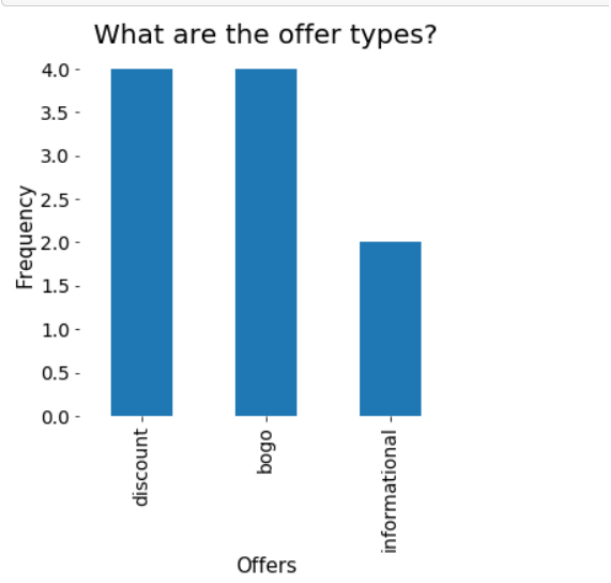
```
models = {'Model': ['KNeighborsClassifier (Benchmark)', b_model, c_model],
          'train F1 score ':[a_train_f1, b_train_f1, c_train_f1],
          'test F1 score': [a_test_f1 , b_test_f1, c_test_f1] }

comp = pd.DataFrame(models)
```

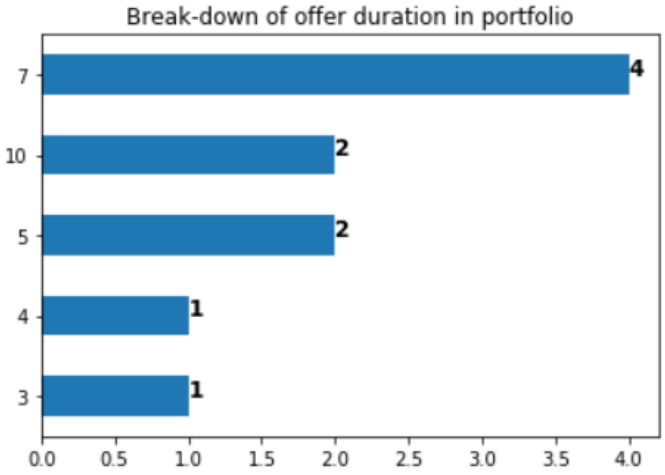
```
comp
```

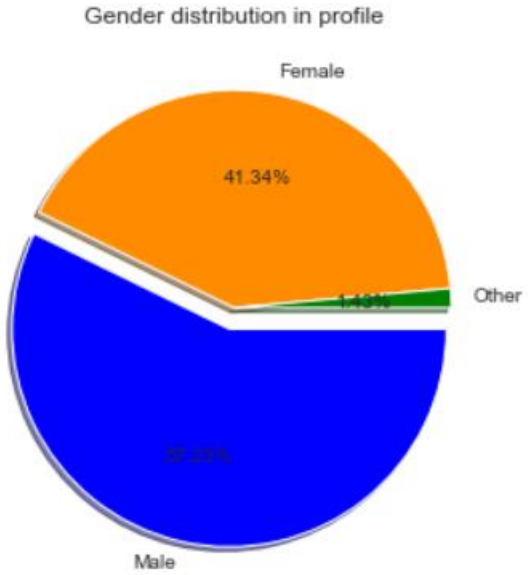
	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	54.346515	32.891019
1	RandomForestClassifier	94.459318	70.702517
2	DecisionTreeClassifier	95.455075	85.079839

The validation set (test data set) is used to evaluate the model. Both the models are better than the benchmark. The best score is created by the DecisionTreeClassifier model, as its validate F1 score is 85.10, which is much higher than the benchmark. The RandomForestClassifier model scores good as well compared to the benchmark, with a test F1 score of 69.30. Our problem to solve is not that sensitive which requires very high F1 score, so the scores are good & sufficient and can be used for the classification purpose to predict whether a customer will respond to an offer.

Data Exploration







Datasets and Inputs

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app.
The data set is provided in form of three JSON files:-
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

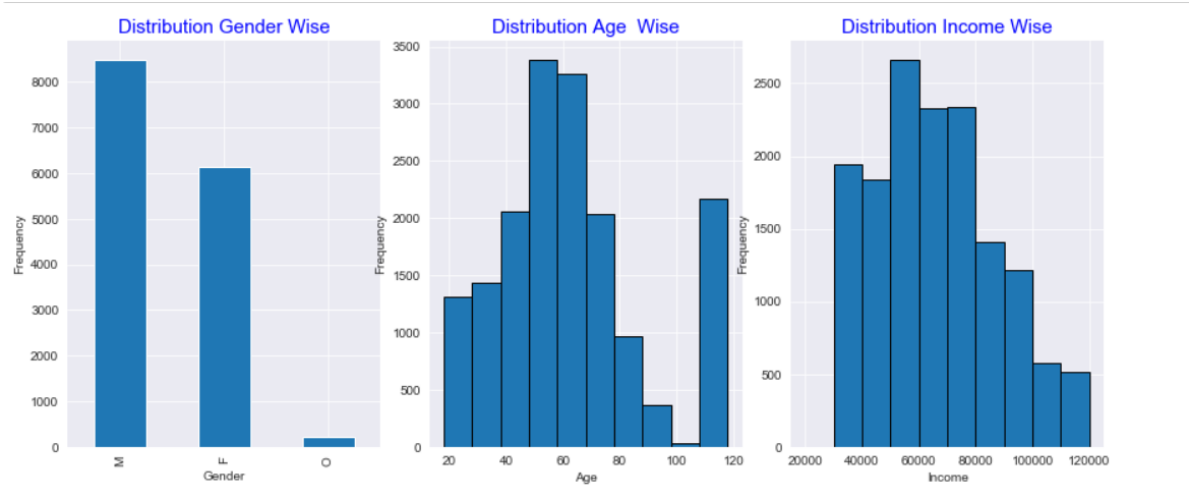
Here is the schema and explanation of each variable in the files:
portfolio.json
•id (string) -offer id
•offer_type (string) -type of offer i.e. BOGO, discount, informational
•difficulty (int) -minimum required spend to complete an offer
•reward (int) -reward given for completing an offer
•duration (int) -time for offer to be open, in days
•channels (list of strings)

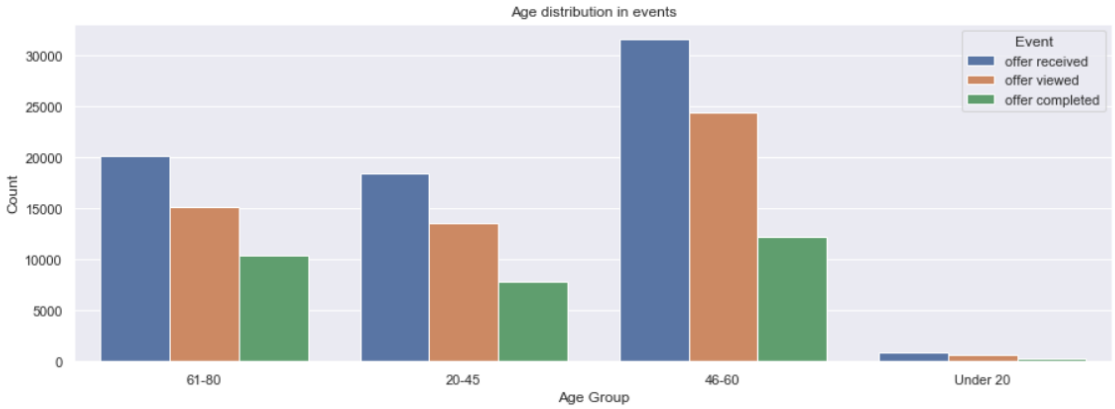
	reward	channels	difficulty	duration	offer_type	id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	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

	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
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN
8	M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
9	None	118	8974fc5686fe429db53ddde067b88302	20161122	NaN

The portfolio.json contains offer_type column, which describes the types of offers that Starbucks is looking to potentially send its customers:
1)BOGO (Buy-One-Get-One): This offer enables a customer to receive an extra and equal product at no additional cost. The customer must spend a certain threshold in order to make this reward available.
2)Informational: This offer doesn't necessarily include a reward, but rather an opportunity for a customer to purchase a certain object given a requisite amount of money.
3)Discount: With this offer, a customer is given a reward that knocks a certain percentage off the original cost of the product they're choosing to purchase, subject to limitations.

Exploratory Visualization





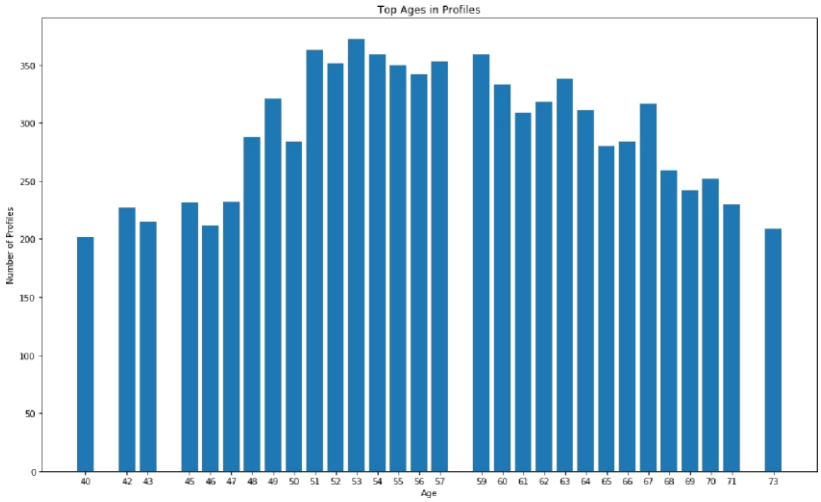
2.2 Exploratory Visualization:

We Will do data visualizing for the data sets before Combination and after combination,we will follow the below Process:

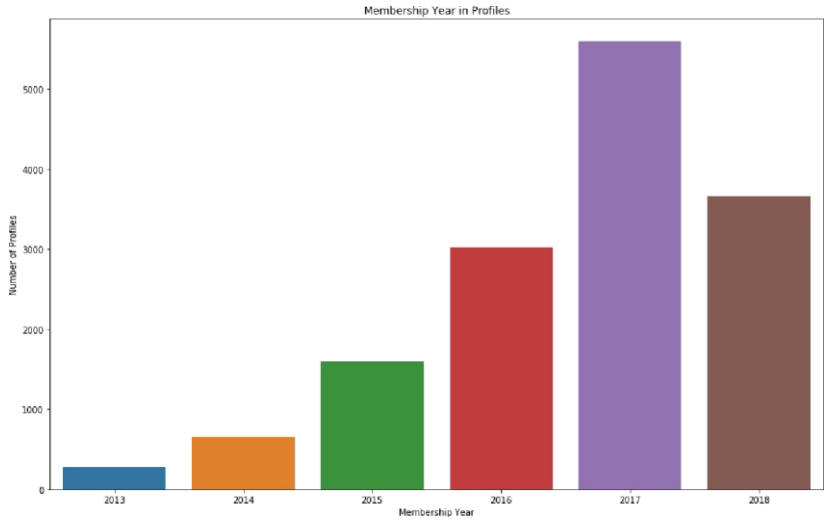
2.2.1 Exploratory Visualization before merging the Data sets.

2.2.1.1 Profile Data Set Exploratory Visualization:

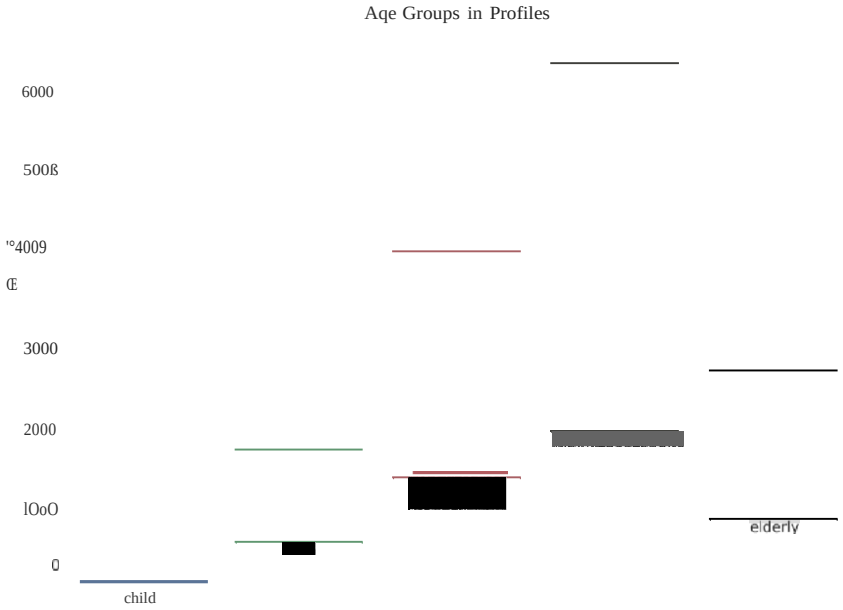
Members age Distribution VS number of Profiles



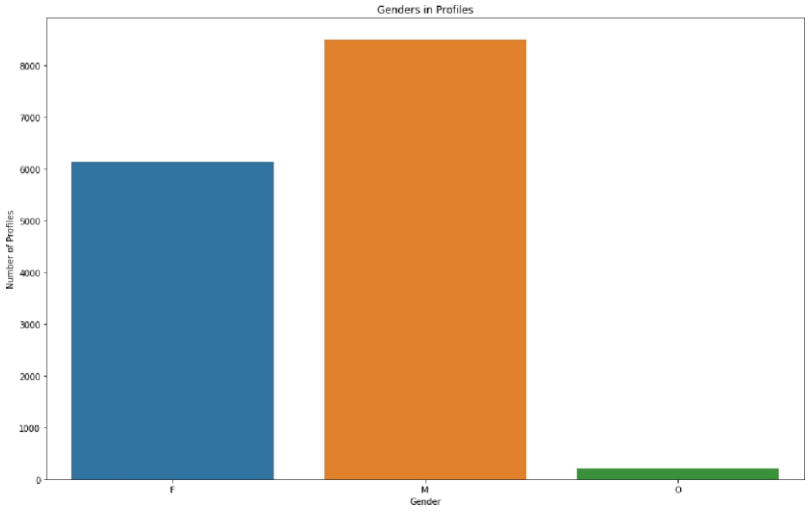
Customers' member ship year VS number of Profiles



Customers' age Groups VS number of Profiles

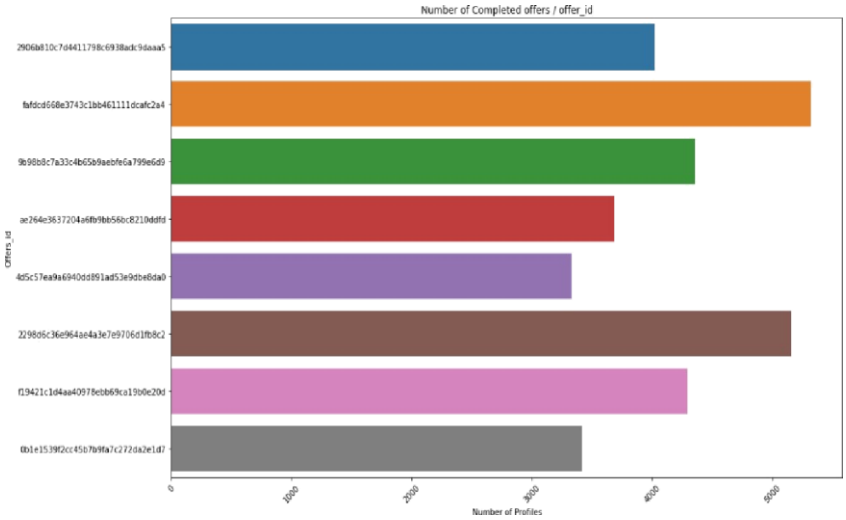


Customers' gender VS number of Profiles

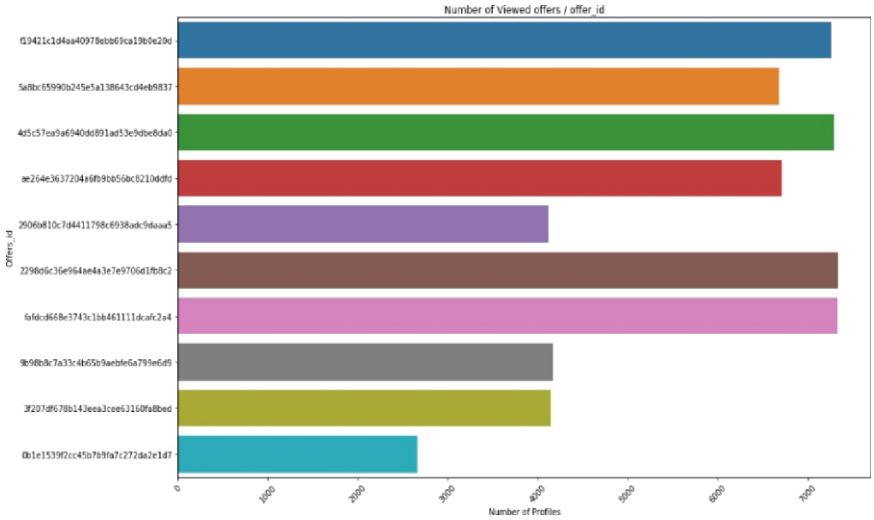


2.2.1.2 Transcript Data Set Exploratory Visualization:

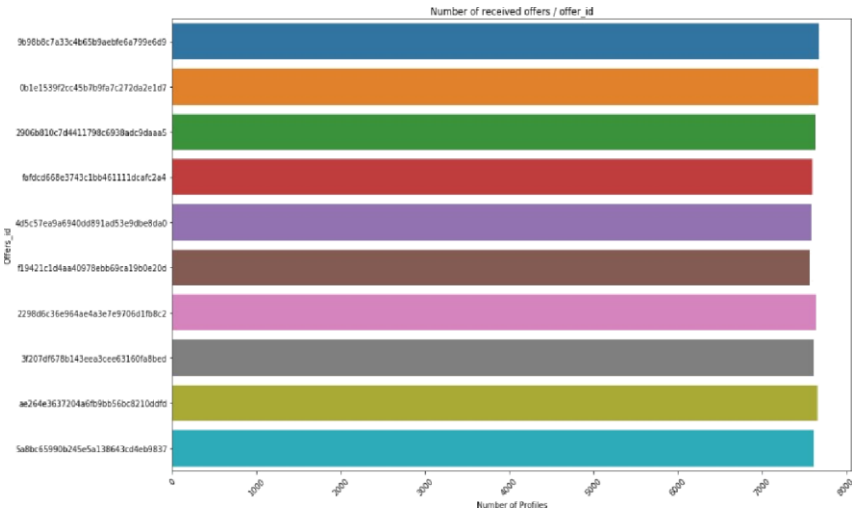
offer id's of Completed offers VS number of Profiles



offer id's of Viewed offers VS number of Profiles

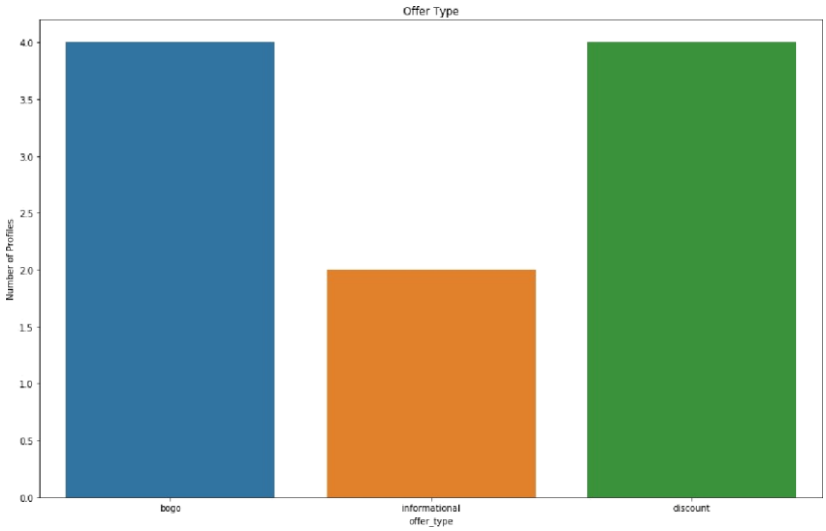


offer id's of Received offers VS number of Profiles



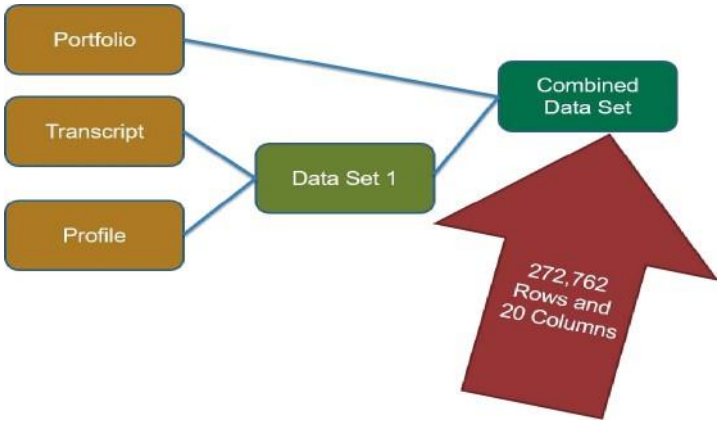
2.2.1.3 Portfolio Data Set Exploratory Visualization

Offers Type VS number of Profiles



2.2.2 Exploratory Visualization after merging the Data sets:

Now we will combine the three data sets, we will follow the below process, then we will do data visualization for the combined data set:



Combining the transcript and profile data set together then Combining the output with Portfoliodata set.

Our output will be the below data frame:

```
Combined_all_data.info()
```

```
offer_id
amount                    272762 non-null object
age
became_member_on          27T762 non-null datetime64[nsj
gender
income

age_groups                27¥V62 con-null categoryyyyv6y
member_launch_Cum_days    non-nu+i int6o z7z2o2  non-null
member_launch_year        int64
difficulty
duration                  insaos non-nuii moat&4
offer_type

                          7488BB non-null object


email
mobile
social
```

Algorithms and Techniques

A quick and fairly accurate model can be considered as a benchmark. I will use the Decision Tree Classifier to build the benchmark, as it is a fast and standard method for binary classification machine learning problems and evaluate the model result usingF1 score as the evaluation metric.

1)Random Forest Classifier: ensemble learning method for classification and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

2) Decision Tree Classifier A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter

3) K-neighbours Classifier A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	54.346515	32.891019
1	RandomForestClassifier	94.459318	70.702517
2	DecisionTreeClassifier	95.455075	85.079839

3. METHODOLOGY

3.1 Data Pre-processing:

3.1.1 Combined Data Preparation for Models training and testing:

A)Dividing our Combined Data to three data sets :1-received: extracting the items with event= offer received. 2-Viewed: extracting the items with event = offer viewed. 3-completed: extracting the items with event = offer completed.4-transaction: extracting the items with event = transaction .

B)(1st output)extracting the persons who completes the received offers ,two new columns to be added to updated data set :-(forecast_finish) column which equals to (received offer time + offer duration) . -(finish) column which equals to (forecast_finish) value and received time value in case of the offer not completed or equals to completion time in case of offer completed. -(completed) column which equals to (1) in case of offer completed and equals to (0) in case of offer not completed.

C)(2nd output) extracting the person who completed the received offer (1st ouput) after viewing the offer within the offer period , three columns to be added-(success) Column which equals to (1) in case of offer completed after viewing the offer other wise equals to (0). -(viewing_time) Column which equals to viewed offer time -(Viewed) Column which equals to either (1) or (0).

D)(3rd output) profits calculation for the amount of money which is spent within the offer forecast completion time assuming that all transaction executed within the offer duration are using the offers Eventually, we will get our Modelled data which will be used in our Models training and testing. We will follow the below Process to get our modelled data starting from Combined data:

Benchmark

We will use Decision Tree Classifier model as a Benchmark in which to compare our models‘s performance to, because it is fast and simple to implement. We will implement the roc_auc_score , Precision and Recall Metrics to Compare other Models ‘s Results.

Hyperparameter Tuning

This is a crucial part of the Machine Learning pipeline. Each algorithm has a set of hyperparameters (different from the parameters, that are calculated at train time) and we have to set them before the training part. For example, for the XGBoost algorithm we can set the number of estimators, the maximum depth for each estimator, the subsample of the train data taken at each step.

3.2 Implementation:

Firstly - after the Preparation of our training and testing data sets -We Will implement our Benchmark model (Logistic regression Model) and calculating our Metrics that we have discussed before

Statistics for **Combined** Data Set:For Females:

Number of offer received offers: 27456 offer, 43.1% of total offers.

Number of offer viewed offers: 20786 offer, 32.6% of total offers.

Number of offer completed offers: 15477 offer, 56.4% of received offers. For Males:

Number of offer received offers: 38129 offer, 46.0% of total offers.

Number of offer viewed offers: 28301 offer, 34.1% of total offers.

Number of offer completed offers: 16466 offer, 43.2% of received offers.

The Maximum value to complete offer for Females: 428.0 Hours and the Value by days is: 17.8 days

The Maximum value to complete offer for Males: 434.0 Hours and the Value by days is: 18.1 days

The Statistics for the offer id's (10 offer ID) VS events type:

Offer ID: 0b1e1539f2cc45b7b9fa7c272da2e1d7 Total number of offers: 12327

Offer ID: 0b1ei539f2cc45b7b9fa7c272da2e1d7 Total number of Completed offers: 3386 and Percentage is: 27.47 %

Offer ID: 0b1e1539f2cc45b7b9fa7c272da2e1d7 Total number of Viewed offers: 2215 and Percentage is: 17.97 %

Offer ID: 0b1e1539f2cc45b7b9fa7c272da2e1d7 Total number of received offers: 6726 andPercentage is: 54.56 %

Offer ID:2298d6c36e964ae4a3e7e9706d1fb8c2 Total number of offers: 17920

Offer ID:2298d6c36e964ae4a3e7e9706d1fb8c2 Total number of Completed offers: 4886 and Percentage is: 27.27 %

Offer ID:2298d6c36e964ae4a3e7e9706d1fb8c2 Total number of Viewed offers: 6379 andPercentage is: 35.60 %

Offer ID:2298d6c36e964ae4a3e7e9706d1fb8c2 Total number of received offers: 6655 andPercentage is: 37.14 %

Offer ID:2906b810c7d4411798c6938adc9daaa5 Total number of offers: 14002

Offer ID:2906b810c7d4411798c6938adc9daaa5 Total number of Completed offers: 3911 and Percentage is: 27.93 %

offer ID:2906b810c7d4411798c6938adc9daaa5 Total number of Viewed offers: 3460 andPercentage is: 24.71%

Offer ID:2906b810c7d4411798c6938adc9daaa5 Total number of received offers: 6631 andPercentage is: 47.35%

Offer ID:3f207df678b143eea3cee63160fa8bed Total number of offers: 10144

Offer ID:3f207df678b143eea3cee63160fa8bed Total number of Viewed offers: 3487 andPercentage is: 34.37 %

Offer ID:3f207df678b143eea3cee63160fa8bed Total number of received offers: 6657 andPercentage is: 65.62 %

Refinement:

In a first time, we did not choose to sample the data: since the unbalance is not so strong, we tried to develop models with the original target distribution. However, we noticed that there were unbalanced predictions: Discount model over-predicted the positive event and the Informational one vice versa. The predictions had the same unbalance as the target. For this reason, we chose to go for under sampling and, as we'll see in the next section, the performances improved quite a lot.

IV. Results

My analysis suggests that the resulting random forest model has an training data accuracy of 0.94 and an F1-score of 0.95. The test data set accuracy of 0.85 and F1-score of 0.80 suggests that the random forest model I constructed did not overfit the training data.

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	54.346515	32.891019
1	RandomForestClassifier	94.459318	70.702517
2	DecisionTreeClassifier	95.455075	85.079839

V. Conclusion

Reflection

Let's summarize all the different steps followed in this process.

1. We analyse the 3 different datasets containing information about offers in Starbucks app

2. Then we reconstruct the customer's journey through offer view, completion and transaction

3. After that, we create new input features to better understand one customer's behavior

4. We made some pre-process on input data

5. We develop different Machine Learning models, comparing them and choosing one champion model for each type of offer I think that wrangling the data and understanding all the different special cases to reconstruct customer' journey was the most difficult and time consuming part.

I personally took several iteration, founding every time some edge case not taken under consideration. Also the model development through the Sagemaker platform was challenging, but very rewarding.

Improvement

We could achieve further improvements in several ways: the most relevant thing could be have more input features. Data about app usage and about which types of products a customer buys could really help understanding one's behavior also having more records at our disposal could improve the model's performances we could set a minimum precision for the Discount model in order to have a non-negative delta with the current

benchmark we could try other algorithms, such as Neural Networks, SVM and Random Forests, as they could better suit this particular use case