Udacity Data Scientist Nanodegree

Starbucks Capstone Project

Jaskaran Singh | 3rd May 2020

Project Overview

Starbucks Corporation is an American coffee company founded in Seattle, Washington in 1971. Starbucks is a passionate purveyor of coffee and other beverages. It ranked 121st in the list of 2019 Fortune 500 Companies. As of early 2019, the company operates over 30,000 locations worldwide.

Like every big corporation out there Starbucks has its mobile app which send various offers to users using the mobile app. An offer is used for advertisement of a drink or a beverage to provide discounts or BOGO (Buy One Get One Free) offers.

The project merely focuses on providing the best promotional offers for customers based on given demographics and response to previous offers to find which offer is most suitable for a particular customer. The company needs to send offers which really benefit the company because sending irrelevant offers is of no use. Also, promotional offers are suitable for those who have less chance of purchasing a beverage/item. There are many platforms which send unnecessary offers and using Machine Learning in such a problem helps the company a lot to make higher profits.

Problem Statement

The problem to solve is to provide the best offers to each user based on their demographics and their response to previous offers. Building a model that predicts whether a customer will respond to an offer or not based on previous response and demographics, so there will be two outputs from the model i.e. A person will either respond or not respond which makes this problem a Binary Classification. But all users will not receive same offer, so solving this problem using the dataset that is provided by Starbucks which is captured for nearly 30 days. Analysing the models based on accuracy and F1 score will help get the best results.

Steps:

1. Data Exploration: Analysing data, its structure, checking for outliers, null value and what we can get out of the given dataset
2. Data Cleaning: Renaming columns, replacing missing values, removing outliers, normalize data
3. Splitting the data: Splitting the data into training and test dataset
4. Training: Training the model with train dataset

5. Testing: Testing the trained model with previously trained model
6. Conclusion: Concluding the outcome for our project

Metrics

The metrics that I have used here is **F1-score** since the data has unbalanced class distribution.

F1-score = 2 * (precision * recall) / (precision + recall)

Analysis

Data Exploration

The data is contained in three files:

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

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

1. 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)

	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

2. profile.json

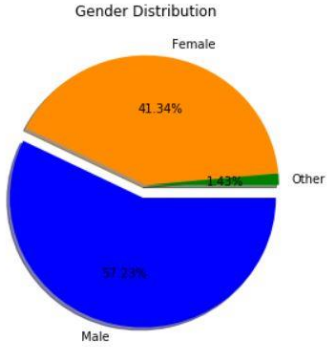
- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

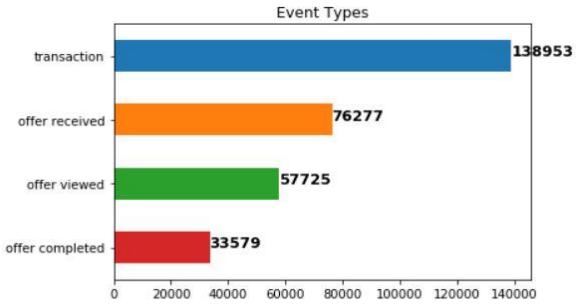
	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN

3. transcript.json

- event (str) - record description (i.e. 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

	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'}





By now we are done with data exploration, now next step will be data cleaning and pre-processing but before let's see the major points we can conclude from data exploration

- The gender and income values are missing for age 118
- People of age more than 80 are not major users of mobile app so act as outliers
- In terms of gender distribution, the no. of males is more than no. of females
- Also, all the offers received are not necessarily completed and some are not even viewed

Data Cleaning and Pre-processing

Cleaning Portfolio

Renaming some columns for better understanding and to make it easier to merge data frames later

Cleaning Profile

We need to do a lot of tasks for Profile data frame and list of all the tasks is given

1. Renaming some columns for better understanding and to make it easier to merge data frames later
2. Replace missing age with mean age value
3. Replace missing income with mean income value
4. Replace missing gender with mode value
5. People with age more than 80 act as outliers so removing them
6. Classifying age into different age groups for better understanding

Cleaning Transcript

The tasks we will perform for the transcript data frame are as follows

1. Renaming some columns for better understanding and to make it easier to merge data frames later
2. Expand the dictionary to columns i.e. Expand the keys of the values column into new columns

Merging the Three Data frames

In this we create a function which takes portfolio, profile and transcript as parameter and returns a merged data frame

Final Cleaning

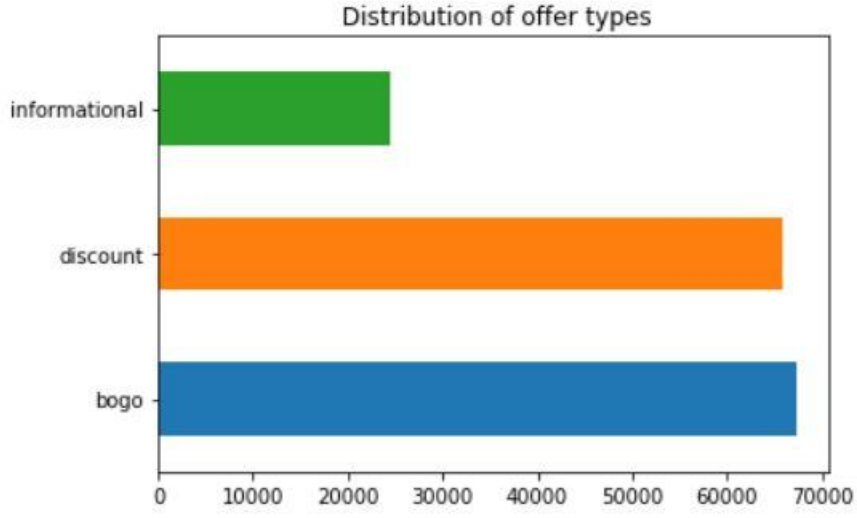
Before building a model, we will have to clean it a bit further. We need to do the following tasks (**NOTE: The below tasks are need to be done after Exploratory Data Analysis**)

1. Convert categorical features into numeric format
2. Encode the 'event' data to numerical values
    - offer received - 1
    - offer viewed - 2
    - offer completed - 3
3. Encode offer_id and customer_id
4. Drop column 'became_member_on' and add separate columns for month and year
5. Scale and normalize numerical data

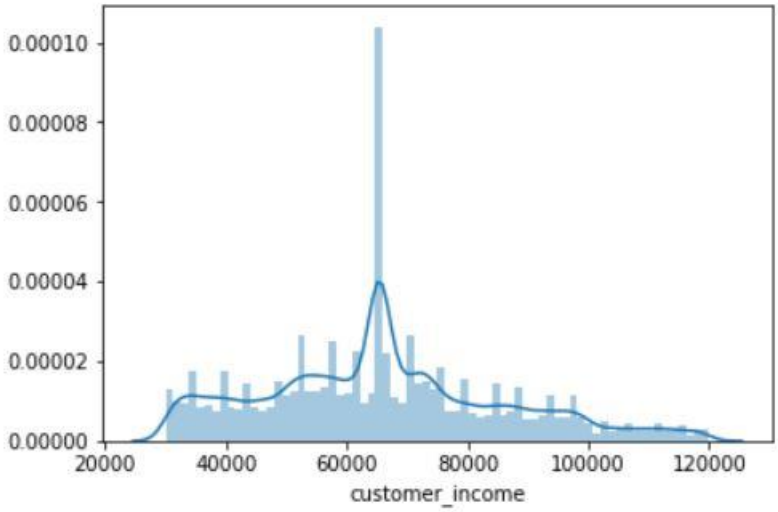
Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations. It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it.

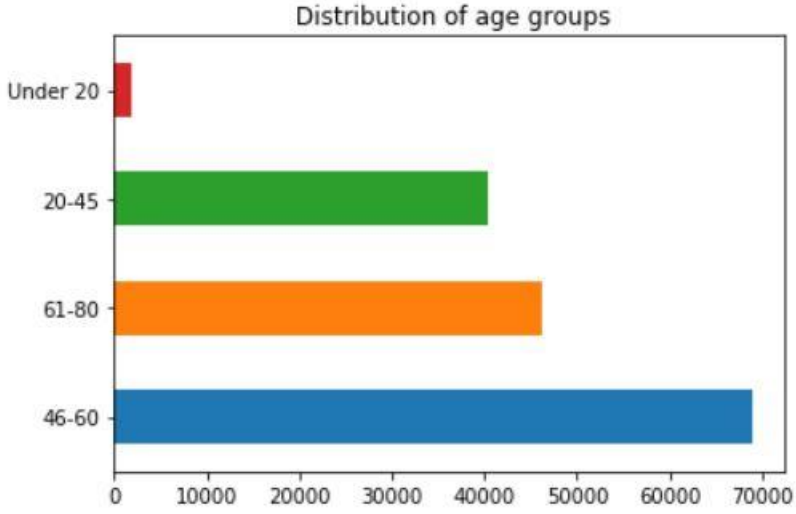
a) <u>Most Used offer by customer using the app</u>



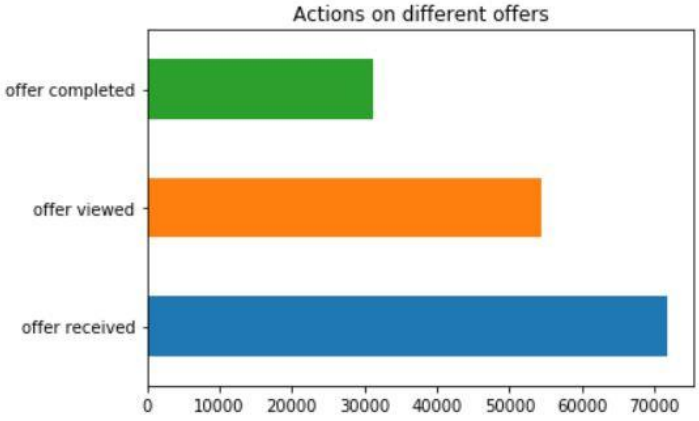
b) Income of customers using the app



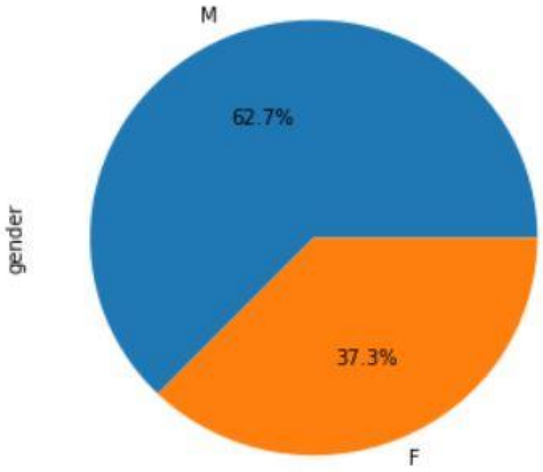
c) Age distribution of customers



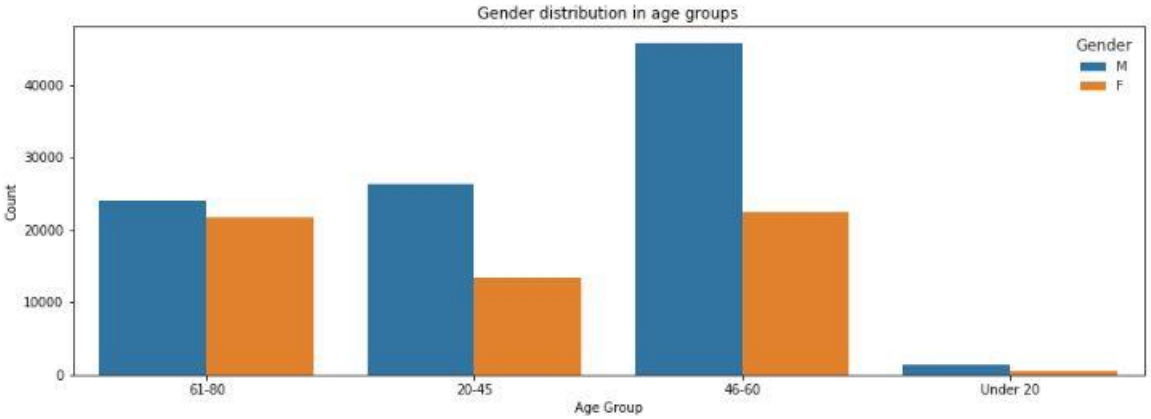
d) Actions to offers that customers received



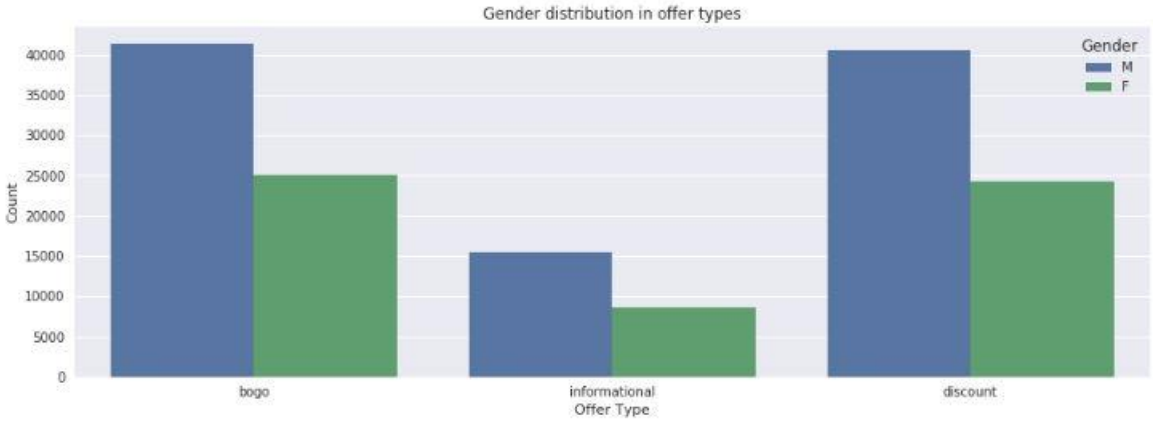
e) Gender based distribution



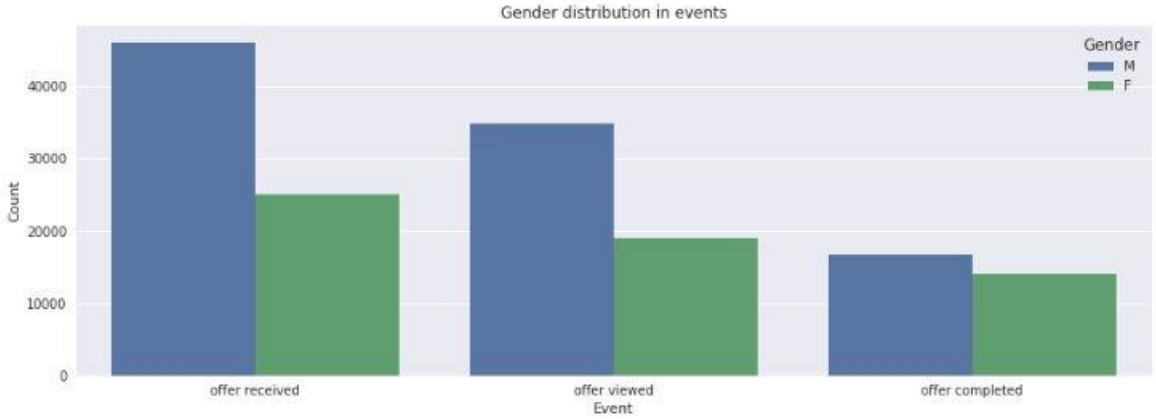
f) Age groups of Males and Females



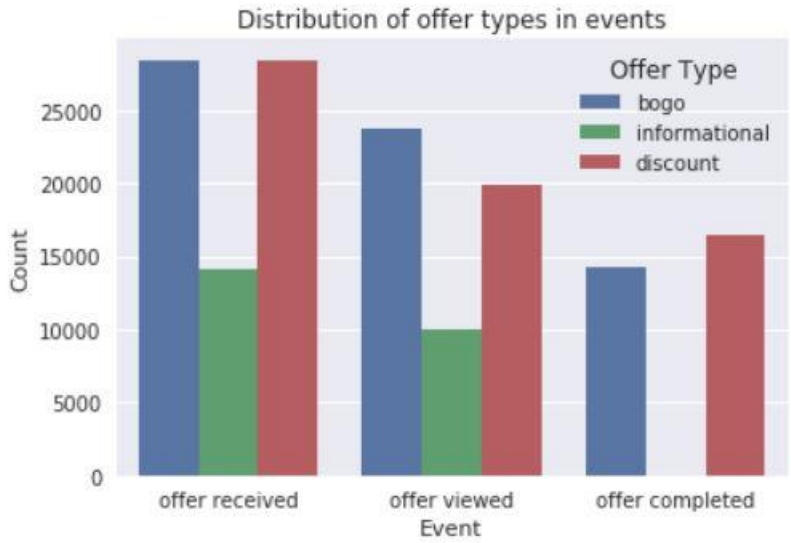
g) Gender distribution in each offer type



h) Action to offers by Male and Females



i) Actions to offers in each offer type



The males represent 62.7% of the data and use the Starbucks app more than the females. Specifically, both males & females in the age group 46-60 use app the most. Also, we can look more to the figures & information in Exploratory Data Analysis to best determine which kind of offers to send to the customers.

Split Train and Test Data

Final data is ready after the last data cleaning tasks. We will now split the data (both features and their labels) into training and test sets, taking 60% of data for training and 40% for testing.

Training and Testing

Evaluation Metrics

We 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 100 and worst at 0.

Training

Benchmark Model:

I will use KNeighboursClassifier to build the benchmark model because it is a fast method for classification problems. As it is quick as well as accurate so it can be considered as a benchmark. Also, it has very easy implementation and we need to select only first hyper parameter and after that rest of parameters are aligned to it.

Other Models Used:

As the problem we have is a classification problem, I used KNeighborsClassifiers, DecisionTreeClassifier and RandomForestClassifier to find the appropriate response of a customer to a particular offer. We are also using RandomForestClassifer because sometimes Decision Trees have a chance of overfitting but we can prevent it using RandomForestClassifier because it's an ensemble classifier which uses many decision trees to predict the result

Evaluation of Models

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	54.346515	32.891019
1	RandomForestClassifier	94.336568	69.304149
2	DecisionTreeClassifier	95.455075	85.098886

The best score is by DecisionTreeClassifier model, as its test F1 score is 85.1 which is much higher than the benchmark

Other Model We Can Try

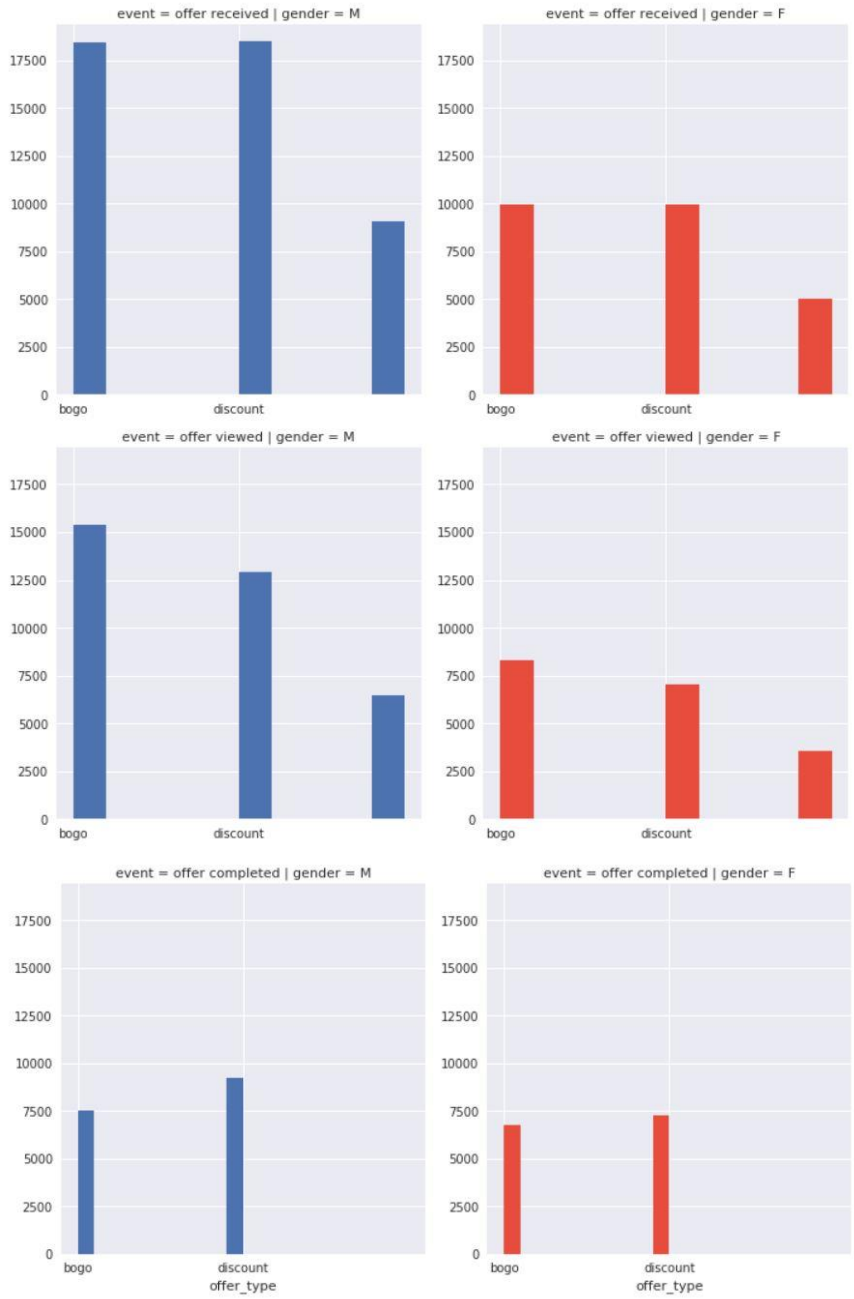
1. GaussianNB
2. AdaBoostClassifier Model
3. SVC

	Benchmark Model	train F1 score	test F1 score
0	GaussianNB	61.778182	62.051998

After trying out these models, the F1 scores were not that great so it was better to go with our decided models RandomForests and DecisionTreeClassifiers. The test F1 score from all these models were in range of 60-65 which is not that good

Conclusion

A) Conclusion From EDA



Major Analysis from Exploratory Data Analysis

1. The males represent 62.7% of the data and use the Starbucks app more than the females.
2. Specifically, both males & females in the age group 46-60 use app the most.
3. Discount offers are more preferred by the customers. Also, there are less number of customers who actually complete the offer as compared to the ones who just view & ignore it.

B) Model Comparisons & evaluation using F1 metric

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	54.346515	32.891019
1	RandomForestClassifier	94.336568	69.304149
2	DecisionTreeClassifier	95.455075	85.098886

We used the test dataset to evaluate the model using F1 metric. We observe that both the model performs better than Benchmark Model (KNeighborsClassifier). The best score is by DecisionTreeClassifier model, as its test F1 score is 85.1 which is much higher than the benchmark. Although RandomForestClassifier scores good compared to benchmark, with a F1 test score 69.30. The F1 scores by DecisionTreeClassifier are good as well as sufficient as our problem to solve is not that sensitive which requires very high F1 score. So, our scores are good and can be used for the classification purpose to predict if a customer will respond to an offer or not

Improvements

The scores that we achieved are good for our classification problem. We can still improve the RandomForestClassifier by refining the hyperparameters using Grid Search with Cross Validation and the DecisionTreeClassifier with K- Fold Cross Validation for hyperparameter tuning.