<u>Udacity Machine Learning Nanodegree Capstone Proposal</u>

Abhimanyu Rajput

Introduction :

It is the Starbucks Capstone Challenge of the Machine Learning Engineer Nanodegree in Udacity. In this project, we will explore the data provided by Starbucks and the given 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 a discount or BOGO (buy one get one free). Some users might not receive any offers during certain weeks.We are going to analyze three file:

portfolio : *containing offer ids and meta data about each offer (duration, type, etc.). 10 rows, 6 columns.*

profile **:** *demographic data for each customer. 17000 rows,5 columns.*

transcript **:** *records for transactions, offers received, offers viewed, and offers completed. 306534 rows, 4 columns.*

The process of our analysis will be by using the CRISP-DM Process (Cross Industry Process for Data Mining) : Define Business understanding, Data understanding,Analyze the data,Modeling the data,Compare model performance and finally selecting one model and improving it.

Business Understanding :

The objective here is to find patterns and show when and where to give specific offer to a specific customer

Data Understanding :

let’s understand data by using tables

First , *portfolio* **:**

```
]: portfolio.head()
```

]:

	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

Second , *profile :*

```
[6]: profile.head()
```

[6]:

	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

Third , *transcript :*

```
[7]: transcript.head()
```

[7]:

	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

Data preparation and wrangling :

For our first dataframe which is portfolio, *we can see that the 'channels' column need some work. Why? because it contains a list, so each value in that list must have its own column. After separating each value, we will obviously need to drop the 'channels' column as it is no longer needed. The table will look like this:*

```
# Now drop the 'channels' column
portfolio = portfolio.drop('channels', axis=1)
portfolio
```

	reward	difficulty	duration	offer_type	id	social	email	web	mobile
0	10	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1	1	0	1
1	10	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	1	1	1	1
2	0	0	4	informational	3f207df678b143eea3cee63160fa8bed	0	1	1	1
3	5	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	1	1	1
4	5	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	1	1	0
5	3	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2	1	1	1	1
6	2	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4	1	1	1	1
7	0	0	3	informational	5a8bc65990b245e5a138643cd4eb9837	1	1	0	1
8	5	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d	1	1	1	1
9	2	10	7	discount	2906b810c7d4411798c6938adc9daaa5	0	1	1	1

Now,

For our next dataframe profile, *we had some NaN and None in both 'gender' and 'income'. First, we will replace the None in 'gender' with N/A then replace the NaN in 'income' with the average of income. After applying these two steps the table will look like this:*

	gender	age	id	became_member_on	income
0	NA	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	65404.991568
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.000000
2	NA	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	65404.991568
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.000000
4	NA	118	a03223e636434f42ac4c3df47e8bac43	20170804	65404.991568
...	...	...	...	...	...
16995	F	45	6d5f3a774f3d4714ab0c092238f3a1d7	20180604	54000.000000
16996	M	61	2cb4f97358b841b9a9773a7aa05a9d77	20180713	72000.000000
16997	M	49	01d26f638c274aa0b965d24cefe3183f	20170126	73000.000000
16998	F	83	9dc1421481194dcd9400aec7c9ae6366	20160307	50000.000000
16999	F	62	e4052622e5ba45a8b96b59aba68cf068	20170722	82000.000000

For our third and final dataframe *transcript,* we can initially confirm that there are no NaN by the following output:

```
transcript.isna().sum()
```

```
person    0
event     0
value     0
time      0
dtype: int64
```

But, as seen above, the 'value' column contains a dictionary that means we have to separate each value and drop the 'value' column as it is no longer needed. To see what value it holds, we use for-loop and find the keys. After iterating through the 'value' column we can find that we have the following keys:['offer id', 'amount', 'offer_id', 'reward']. Our next step is to iterate over transcript table, check value column and update it, put each key in separated column, and finally delete the 'value' column. After applying what I've discussed, the table will look like this:

```
transcript = transcript.drop('value', axis=1)
transcript.head()
```

	person	event	time	offer_id	amount	reward
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	0	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	0	0
2	e2127556f4f64592b11af22de27a7932	offer received	0	2906b810c7d4411798c6938adc9daaa5	0	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	0	fafdcd668e3743c1bb461111dcafc2a4	0	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	0	4d5c57ea9a6940dd891ad53e9dbe8da0	0	0

Looks Good!

Everything looks setup for Analysis and Modeling

Analyzing the Data :

*For this part, it will be divided into **Univariate Exploration** and **Multivariate Exploration.***

*First, let's start with the **Univariate Exploration** and try to answer the following questions:*

1. **We will start with the first question, what is the average income for Starbucks customers? its quite easy to calculate simply use the *.mean()* and the average income is:**

```
profile['income'].mean()
```

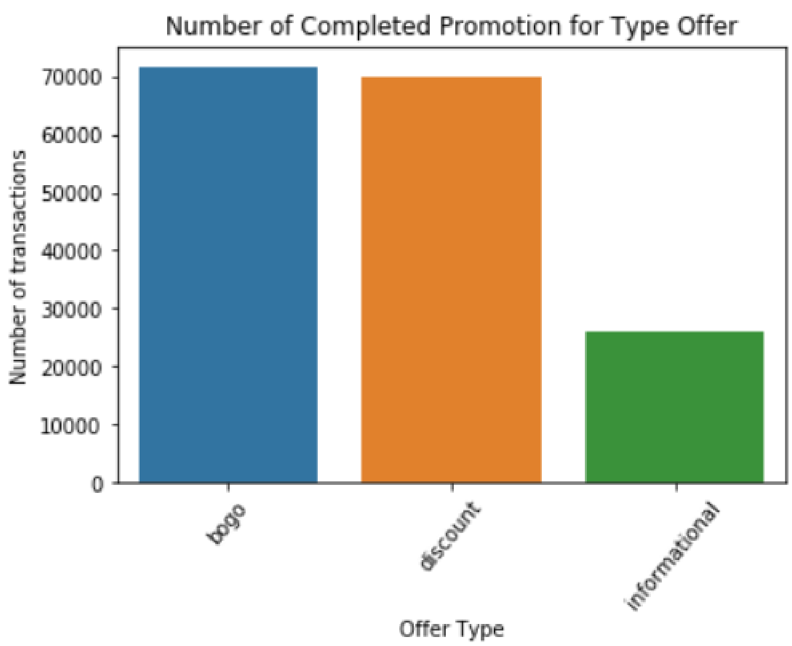
```
65404.99156829799
```

2. **On to the next question. What is the average age for Starbucks customers? to do this it is similar to our previous question, use the *.mean()* function. The output is:**

```
profile['age'].mean()
```

```
62.53141176470588
```

3. **Our third question is, What is the most common promotion? This one was a little bit tricky as they needed to be converted to text first. After converting and encoding/decoding, I decided to show the top 3 promotion only and show only the completed promotions as they are more important. So, we got the following output:**

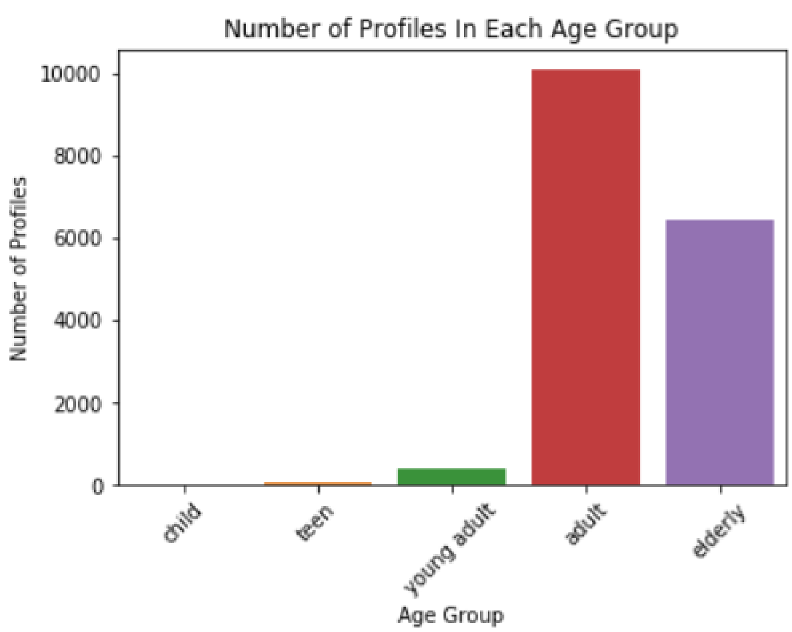


WOW! *BOGO*(buy one get one free) is the most used followed be *discount* with a small difference. While *informational* came third with ~40000 difference, that's a huge gap.

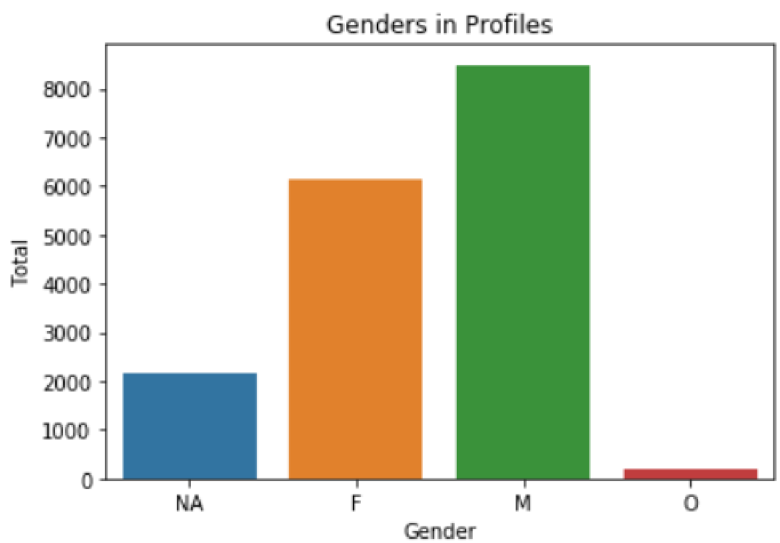
4. On to the forth questions, what are the most common age group and gender? First, I decided to use age group to make it easier to deal with and read. The following code shows how I divided them:

```
profile['age_groups'] = pd.cut(profile.age, bins=[0, 12, 18, 21, 64, 200],
                               labels=['child', 'teen', 'young adult', 'adult', 'elderly'])
```

After creating the *age_group,* we can start answering our question. Let's look at what age group most customers are:



Now let’s check the *gender* groups:



WOW! most are Males with ~2000 difference from the closet which is Females.

5. **Now let’s look at our final Univariate related question, Who are the most loyal customer (most transcripts)? This might help us so we can give them more promotion to rewards their loyalty 😊. To approach this question, we can order the ‘amount’ in descending order and get the top 10. After applying the action, we will get the following output :**

```
.------------------ [ #1 ] ------------------.
| Profile ID: 3c8d541112a74af99e88abbd0692f00e |
| Number of Completed Offers:       5          |
| Amount:                         $1606        |
'----------------------------------------------'
.------------------ [ #2 ] ------------------.
| Profile ID: f1d65ae63f174b8f80fa063adcaa63b7 |
| Number of Completed Offers:       6          |
| Amount:                         $1360        |
'----------------------------------------------'
.------------------ [ #3 ] ------------------.
| Profile ID: ae6f43089b674728a50b8727252d3305 |
| Number of Completed Offers:       3          |
| Amount:                         $1320        |
'----------------------------------------------'
.------------------ [ #4 ] ------------------.
| Profile ID: 626df8678e2a4953b9098246418c9cfa |
| Number of Completed Offers:       4          |
| Amount:                         $1314        |
'----------------------------------------------'
.------------------ [ #5 ] ------------------.
| Profile ID: 73afdeca19e349b98f09e928644610f8 |
| Number of Completed Offers:       5          |
| Amount:                         $1314        |
'----------------------------------------------'
```

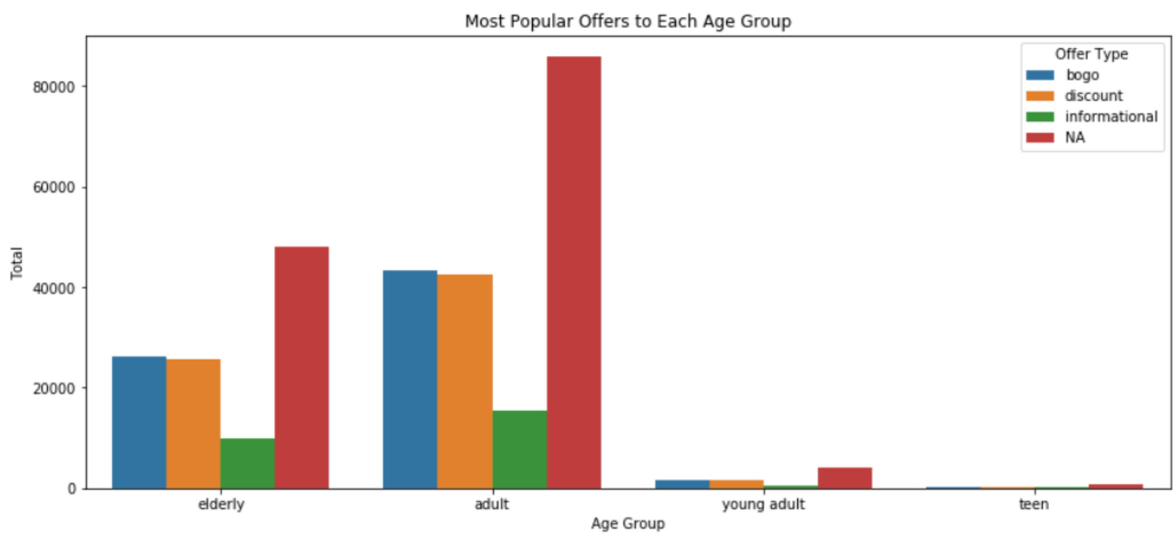
```
.------------------ [ #6 ] -----------------.
| Profile ID: 52959f19113e4241a8cb3bef486c6412 |
| Number of Completed Offers:       5          |
| Amount:                         $1285        |
'----------------------------------------------'
.------------------ [ #7 ] -----------------.
| Profile ID: ad1f0a409ae642bc9a43f31f56c130fc |
| Number of Completed Offers:       3          |
| Amount:                         $1256        |
'----------------------------------------------'
.------------------ [ #8 ] -----------------.
| Profile ID: d240308de0ee4cf8bb6072816268582b |
| Number of Completed Offers:       5          |
| Amount:                         $1244        |
'----------------------------------------------'
.------------------ [ #9 ] -----------------.
| Profile ID: 946fc0d3ecc4492aa4cc06cf6b1492c3 |
| Number of Completed Offers:       4          |
| Amount:                         $1224        |
'----------------------------------------------'
.------------------ [ #10 ] -----------------.
| Profile ID: 6406abad8e2c4b8584e4f68003de148d |
| Number of Completed Offers:       3          |
| Amount:                         $1206        |
'----------------------------------------------'
```

As shown above, we can see their Profile ID as each customer has a unique number, Number of Completed Offers, and the Amount. By this data, we can give them extra and unique promotions in order to reward them.

Now that we have completed the Univariate part, let's move move on to next one.

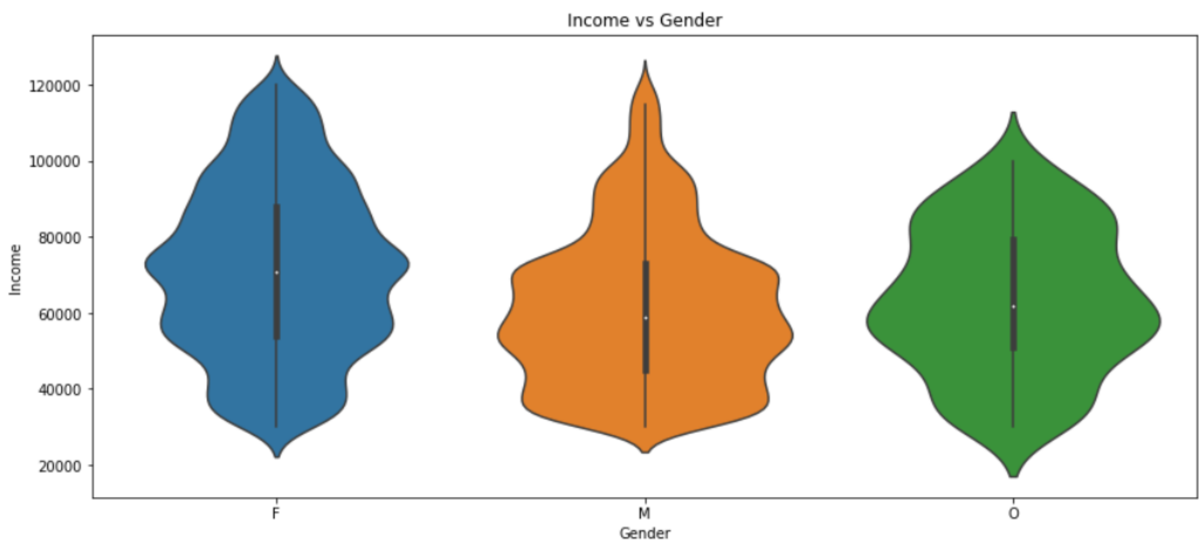
For our **Multivariate Exploration,** we will try to answer the following questions:

1. **Let's look at the first question, what is the most common promotion for children, teens, young adult, adult and elderly customers? Since it's a Multivariate question we'll use a multi bar chart. The output will be:**



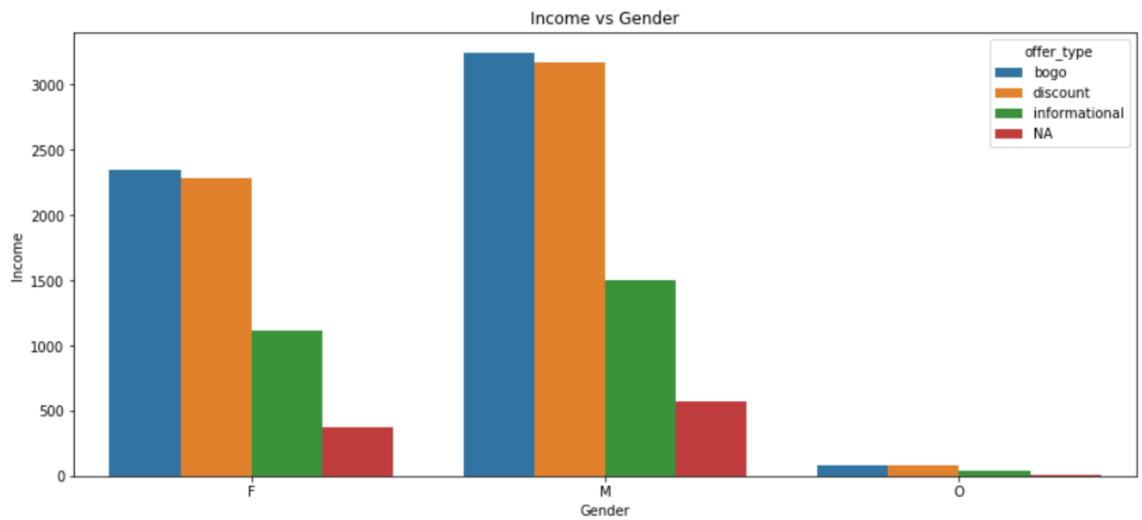
We can observe that all of them have similar results in offer type, Transactions has the upper hand, followed by BOGO. We can also see that young adults and teens aren't our main customer group, so we can focus on elderly and adults.

2. **Our second questions is, from profiles, which get more income, males or females? For this question we will ignore the N/A group because they haven't specified their gender. Furthermore, we will use a plot called violin and use both *income* and *gender* to answer our question. The output will look like:**



The graph above shows that income median (the white dot) for females (around 70k) is higher than males (around 60k) we can also see that for females the income spreads from 40k to 100k. For males most of them around 40k to 70k which close to median.

3. **Our third and final question is, which type of promotions each gender likes? This questions is similar to our first question, but our focus now is on gender. So, we’ll use a multi bar chart.**



It seems that they all share the same interest and prefer BOGO 😅, but we can’t ignore discount and the difference between them is low.

It looks like we have successfully covered the analysis part. Now, we will focus and machine learning and applying different models.

Modeling the Data :

I tried to make a model that can identify which kind of offers we should give a customer. Because my model will guess the offer_type, I will only get those transcripts with offer id's. So I will ignore all transactions without offer id's.

Since we have a simple classification problem, I will use accuracy to evaluate my models. We want to see how well our model by seeing the number of correct predictions vs total number of predictions. Why choose accuracy? First let's define accuracy, the ratio of the correctly labeled subjects to the whole pool of subjects. Also, accuracy answers questions like: How many students did we correctly label out of all the students? It's similar to our situation right? because we want to see how many customers use Starbucks offers. Furthermore, Accuracy = (TP+TN)/(TP+FP+FN+TN). Not to forget, that this is a simple classification problem, so this is my opinion and reasoning on Our features will be:why to use the easiest (accuracy).

Our features will be :

- Event. (Will be replaced from categorical to numerical)
- Time. (normalized)
- Offer_id. (Will be replaced from categorical to numerical)
- Amount. (normalized)
- Reward. (normalized)
- Age_group. (Will be replaced from categorical to numerical)
- Gender. (Will be replaced from categorical to numerical).
- Income. (normalized)

While our target will be offer type.

The models that I have used are: Logistic Regression, K-Nearest Neighbors, Decision Tree, Support Vector Machine, Random Forest, and Naive Bayes.

Compare model performance :

Now that we have trained the data, it’s time to evaluate their performance based on accuracy.

	LogisticRegression	KNeighborsClassifier	DecisionTreeClassifier	SVC	RandomForestRegressor	GaussianNB
Training Accuracy	80.526316	99.999565	100.0	100.0	100.0	72.441931
Predicting Accuracy	92.810000	100.000000	100.0	100.0	100.0	78.730000

Based on the above table, we can see that we’ve scored 100% accuracy in the training and testing datasets on 4 models. To avoid overfitting, I will choose *Logistic Regression* since it got good results 80.5% on training and 92.8% on testing datasets. *Logistic Regression* is better used here since we have few binomial outcomes. It also good here because we have a decent amount of data to work with. Now, let’s improve our model to have better results.

Model Improvements :

After using *Grid Search* with *Logistic Regression* we managed to get better results as shown here :

```
Best Score: 0.8236842105263158
Best params: {'C': 4.0, 'dual': True, 'max_iter': 120}
```

Almost a 1.7% increase, which is great! I don’t think it needs further improvements.

Conclusion :

In this project, I tried to analyze and make model to predict the best offer to give a Starbucks customer. First I explored the data and see what I have to change before start the analysis. Then I did some exploratory analysis on the data after cleaning. After that I trained the data, then choose one model and improved it to get better results. In conclusion, I think that Starbucks needs to focus more on adults and Males.Also, offer more BOGO and discounts to their customers.