Marketing Promotion Campaign Uplift Modelling [Starbucks Dataset]

(Customer Retention data for Churn Prediction or Uplift Modelling)

Aim:

- Given the details of a customer, predict if he falls into the persuadable category or not
- Analyze the results of the experiment and identify the effect of the Treatment on product purchase and Net Incremental Revenue
- Build a model to select the best customers to target that maximizes the Incremental Response Rate and Net Incremental Revenue.

Theory:

A randomized experiment was conducted on Starbucks consumer dataset.

Treatment – Indicates if the customer was part of treatment or control

Purchase – Indicates if the customer purchased the product

ID – Customer ID

V1 to V7 – features of the customer

Cost of sending a Promotion: $0.15

Revenue from purchase of product: $10 (There is only one product)

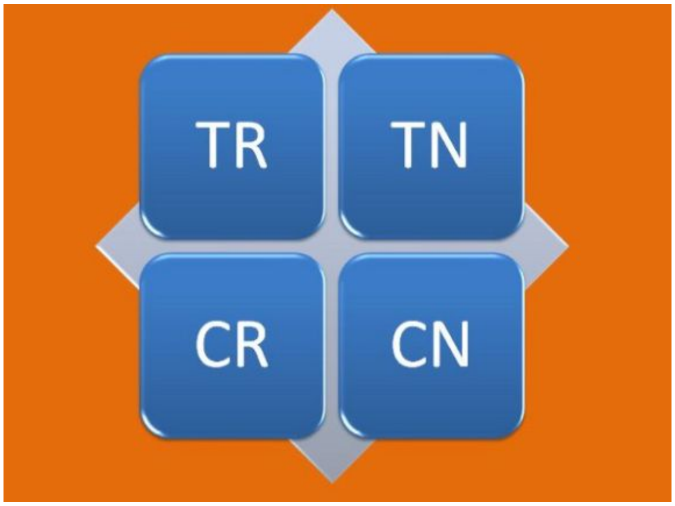
This dataset concerns an experiment involving a promotional campaign. As part of the experiment, some customers were given promotions to entice them to purchase a product. Each product has a purchase price of $10, and the cost of each promotion is $0.15. Ideally, it would be best to limit that promotion only to those that are most receptive to the promotion.

Any customers who received a promotion will be classified as belonging to the treatment group. A second group of customers, known as the control group, were not given the promotions. In both groups, the dataset also tracked if the customers ended up purchasing the product or not.

In addition, seven additional unnamed features, V1 … V7, associated with each data point were provided as well.

We have followed Four Quadrant Approach here.

This approach utilizes a single model to predict the probability of an individual belonging to one of the 4 categories (quadrants), and hence its name. These categories are:



The four quadrant approach predicts the probability that an individual will belong to one of the 4 classes

1. TR: the treatment and response group. Individuals in this group received a treatment (promotion) and responded (made a purchase)
2. CR: the control and response group. Individuals in this group received no treatment (no promotion) but still responded (made a purchase)
3. TN: the treatment and no response group. Individuals in this group received a treatment (promotion) but did not respond (made no purchase)
4. CN: the control and no response group. Individuals in this group received no treatment (no promotion) and did not respond (made no purchase)

We have separated the training data into the appropriate groups and assigned the correct labels to them.

If a model predicts that an individual belongs to class TR, it is likely that he or she will respond favorably to the promotion and we should send a promotion to that individual. If other classes are predicted, we should not send a promotion.

We have used the following key metric to track the performance of our model:

Incremental Response Rate (IRR):

IRR measures how many more customers purchased the product with the promotion, as compared to if they didn't receive the promotion.

$$IRR = \frac{purch_{treat}}{cust_{treat}} - \frac{purch_{ctrl}}{cust_{ctrl}}$$

Net Incremental Revenue (NIR):

NIR measures how much is made (or lost) by sending out the promotion.

$$NIR = (10 \cdot purch_{treat} - 0.15 \cdot cust_{treat}) - 10 \cdot purch_{ctrl}$$

Technical aspects:

Firstly, since we're using the four quadrant approach, we need to create a column with appropriate target values. We use the following notation:

0 - TR group (Treatment and Response group) : Persuadables
1 - CR group (Control and Response group) : The sure shots
2 - TN group (Treatment and No response group) : The Do Not Disturbs
3 - CN group (Control and No response group) : The lost causes

```
target = []
for index, row in train_data.iterrows():
    if (row['Promotion'] == "Yes") & (row['purchase'] == 1):
        # TR group (Treatment and Response group) --> Persuadables
        target.append(0)
    elif (row['Promotion'] == "No") & (row['purchase'] == 1):
        # CR group (Control and Response group) --> The sure shots
```

```
        target.append(1)
    elif (row['Promotion'] == "Yes") & (row['purchase'] == 0):
        # TN group (Treatment and No response group) --> The Do Not
Disturbs
        target.append(2)
    else: #CN group (Control and No response group) --> The lost causes
        target.append(3)

train_data['target'] = target
```

The target column is what we need to predict once our model is trained.

If we take a look at the “target” column -

Total no.of persuadables (TN group [0]) = 580
Total no.of do not disturbs (CR group [1]) = 252
Total no.of sure things (TN group [2]) = 33351
Total no.of lost causes (CN group [3]) = 33444

This shows that the dataset is highly imbalanced and we need a technique to handle such an imbalance. Failing to do so would result in our Machine Learning model to predict everyone to have not bought the product.

We will be using **SMOTE (Synthetic Minority Oversampling Technique)** for handling such an imbalance i.e. We will be up-sampling the minority classes in the training data with SMOTE. Up-sampling would result in an equal number of 0s, 1s, 2s and 3s (number of 0s, 1s and 2s getting increased to match the number of 3s) in the conversion column while generating dummy values for the rest of the columns which mimic the real data that we have. Note that we’ll be performing up-sampling only on our training data.

```
# up sample with SMOTE
sm = SMOTE({0:33444, 1:33444, 2:33444, 3:33444}, random_state=42)
X_train_upsamp, Y_train_upsamp = sm.fit_sample(X_train, Y_train)

X_train_upsamp = pd.DataFrame(X_train_upsamp, columns=features)

Y_train_upsamp = pd.Series(Y_train_upsamp)
Y_train_upsamp.value_counts()
```

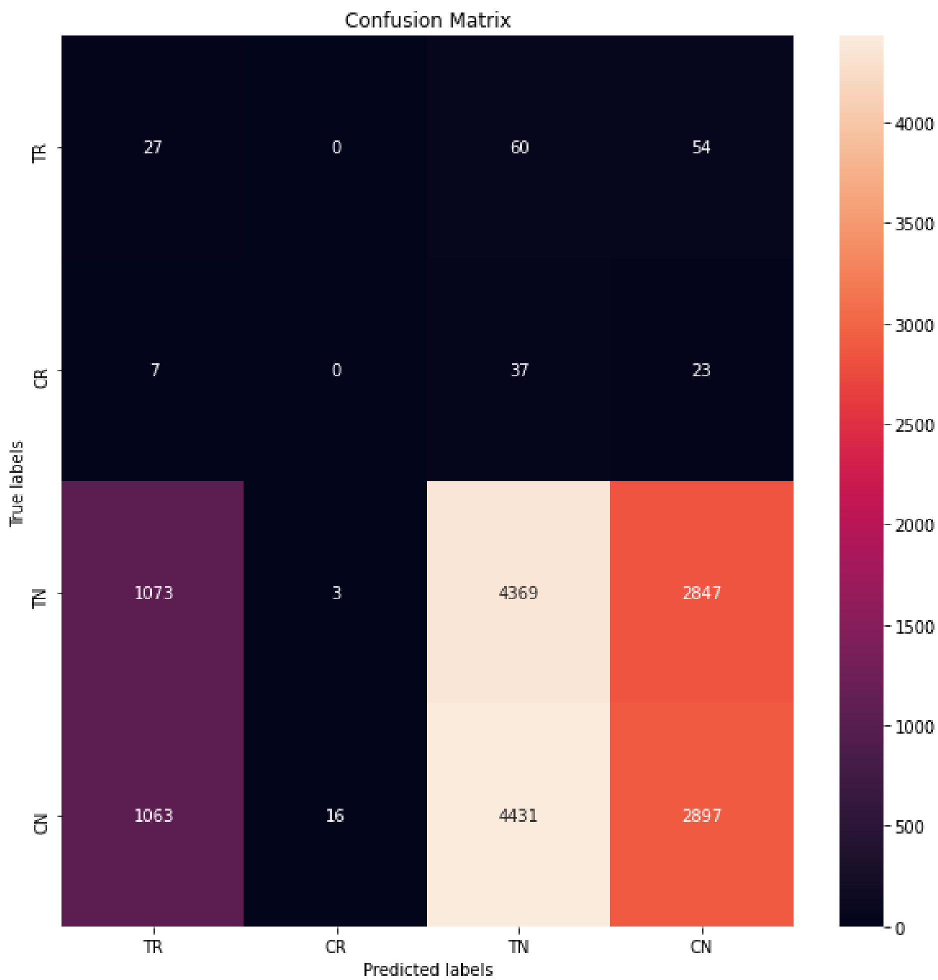
Y_train_upsamp now has 33444 0s, 1s, 2s and 3s.

We have used the XGBoost algorithm to train our Machine Learning model. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Presented in a research paper in 2016, this algorithm is considered the best in class for small-to-medium structured/tabular data.

```
eval_set = [(X_train_upsamp, Y_train_upsamp), (X_valid, Y_valid)]
model = xgb.XGBClassifier()
model.fit(X_train_upsamp, Y_train_upsamp, eval_set=eval_set,\
                     eval_metric="mlogloss", verbose=True,
early_stopping_rounds=30)
```

Now our model is ready to make predictions. If the model predicts a label of 0 for that individual, then it is likely that the individual will respond favorably to the promotion campaign and we should send him or her the promotion. Otherwise, we should not send a promotion.

We obtained the following results:

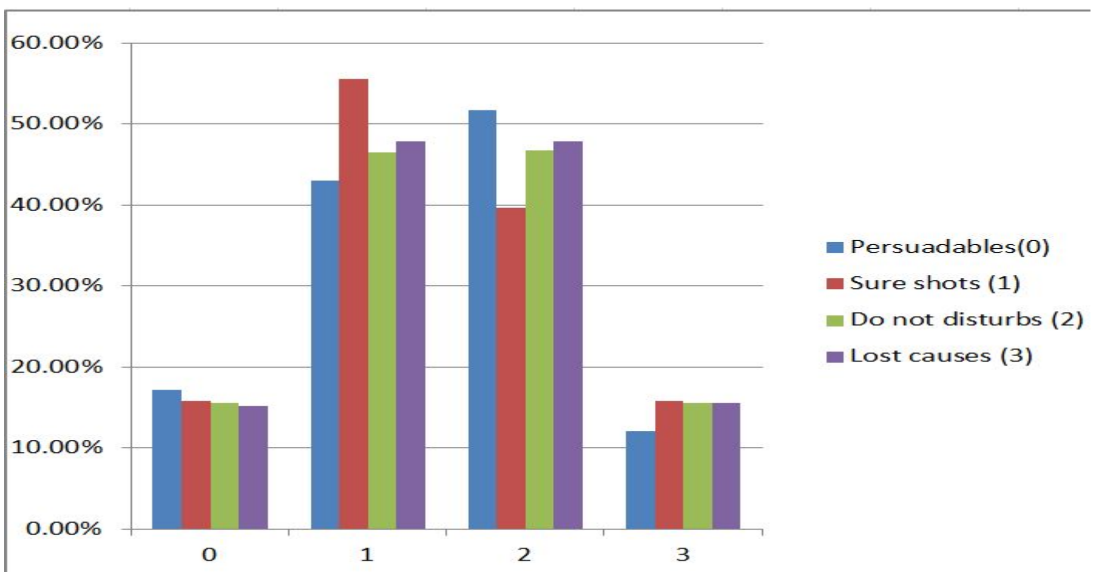


This model achieved an **IRR of 2.36%** on the test set which is 0.48% more than what the starbuck's baseline model predicted (1.88%) and an NIR of 214.95$ which is

Graph Analysis:

Total count of dataset values in training set= 84,534
Total count of dataset values in test set= 41,6450
Total no.of persuadables=721
Total no.of do not disturbs=319
Total no.of sure things= 41,643
Total no.of lost causes=41,851

V1



In V1,

0- There is no considerable difference amongst the 4 categories. Hence, 0 won't be an ideal value in V1 to segregate the consumers.

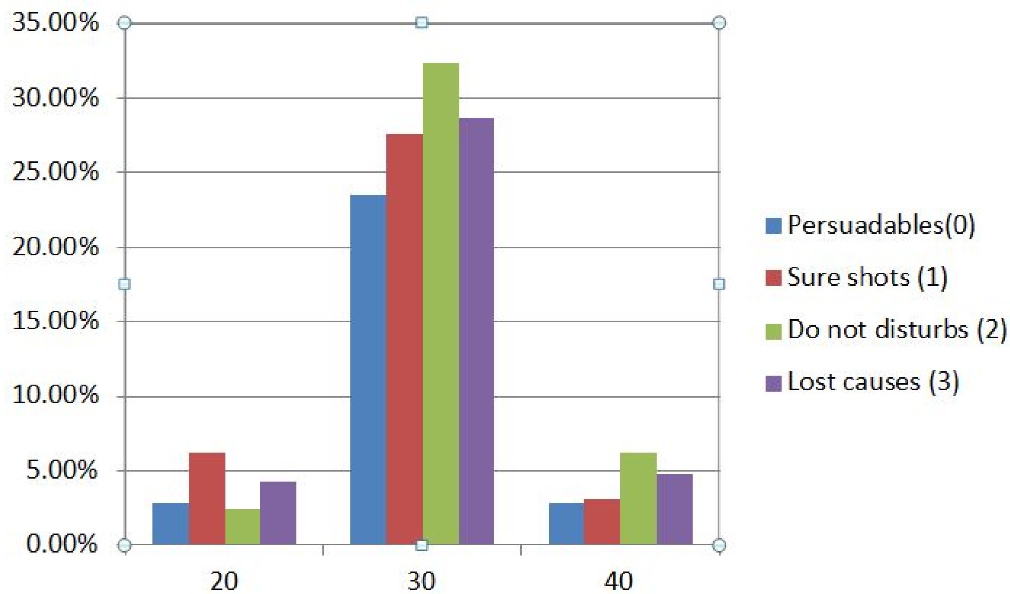
1- There is a considerable difference in the graph of persuadables of more than 5% from the other 3 categories. Hence, 1 will be an ideal value to separate persuadables in V1.

2- There is a considerable difference in the graph of persuadables of more than 4% from the other 3 categories. Hence, 2 will be an ideal value to separate persuadables in V1, but less ideal than 1.

3- There is only a difference of 3% or less difference amongst persuadables and the other 3 categories. Hence, 3 won't be an ideal value in V1 to segregate the consumers.

Result- In V1, 1 would be the most ideal value to distinguish persuadables from the other 3 categories.

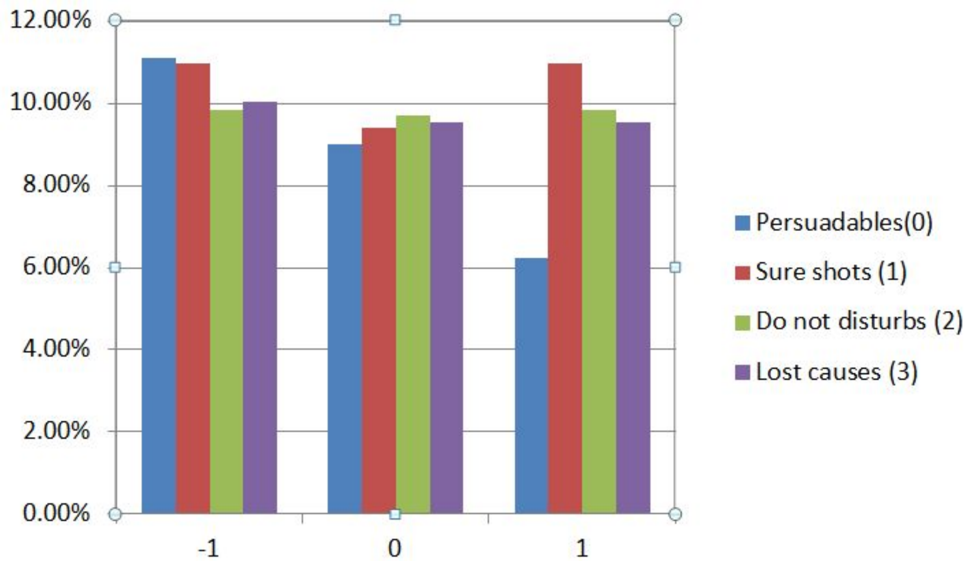
V2



In V2,

With 30 on the x-axis, the graph of persuadables has a difference of 7% and more from the rest of the 3 consumers.
Result- 30 in V2 is the most ideal value to distinguish persuadables from the rest of the 3 consumers.

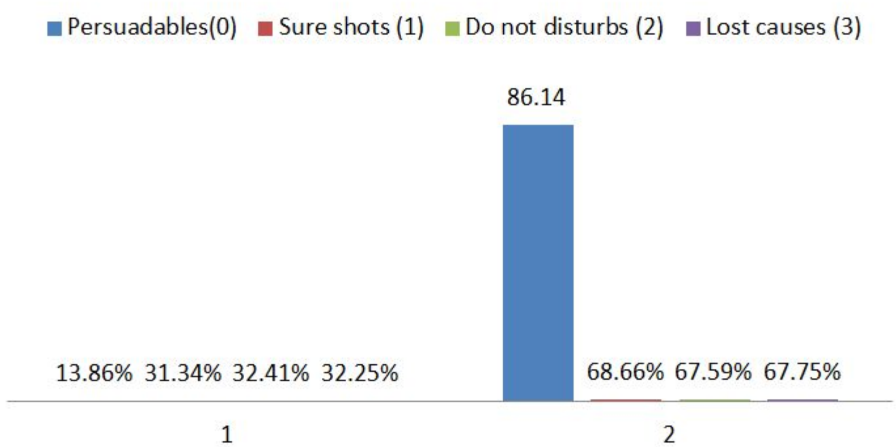
V3



In V3,

With 1 on the x-axis, the graph of persuadables has a difference of 4% and more from the rest of the 3 consumers.
Result- 1 in V3 is the most ideal value to distinguish persuadables from the rest of the 3 categories.

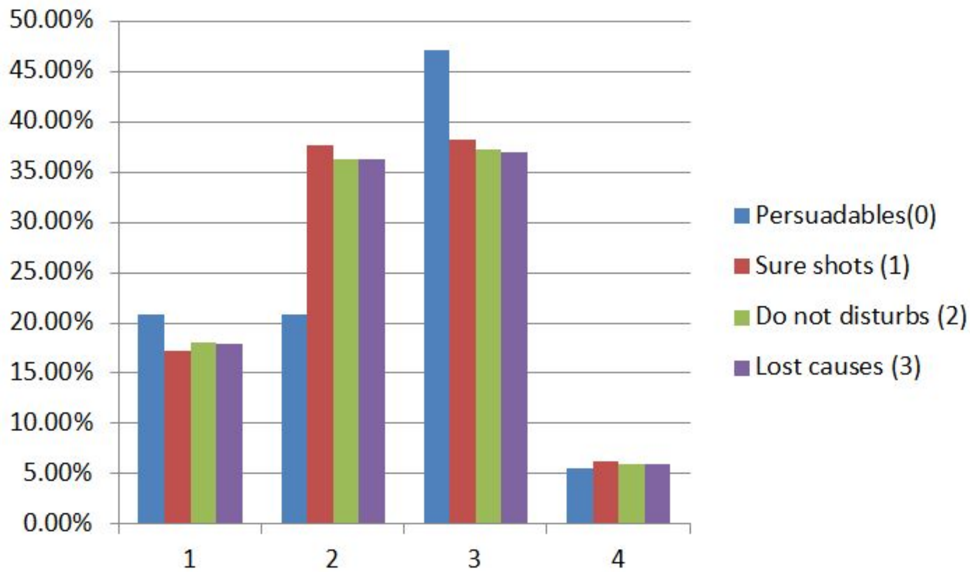
V4



In V4,

With 2 on the x-axis, the graph of persuadables has the tallest bar.
Result- 2 in V4 is the most ideal value to distinguish persuadables from the rest of the 3 categories.

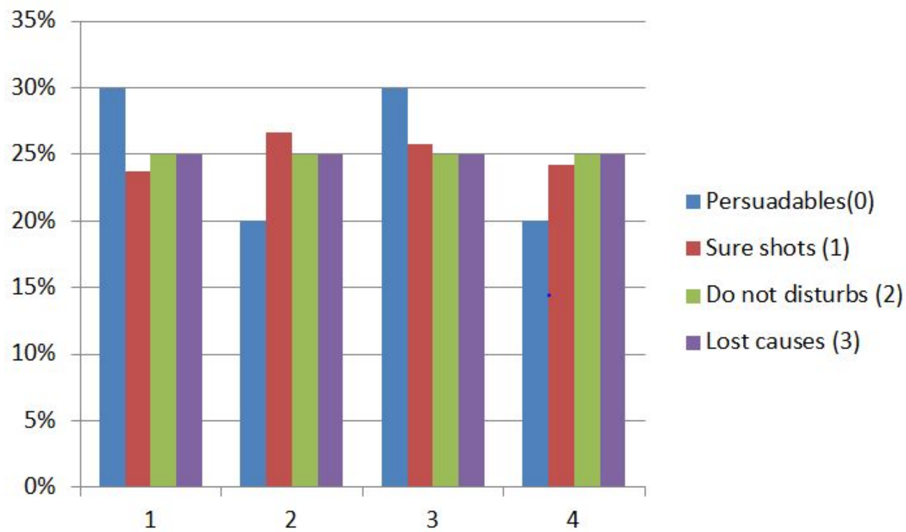
V5



In V5,

With 2 on the x-axis, the graph of persuadables has the shortest bar.
Result- 2 in V5 is the most ideal value to distinguish persuadables from the rest of the 3 categories.

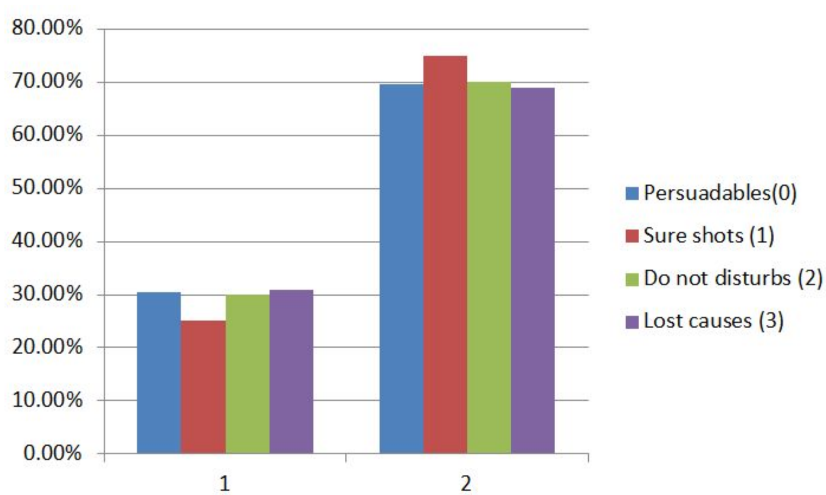
V6



In V6,

With 1, 2, 3 and 4 on the x-axis, the graph of persuadables has considerable differences. Result - All the values on the x axis, 1, 2, 3 and 4 in V6 are ideal values to distinguish persuadables from the rest of the 3 categories.

V7



In V7,

Out of 1 and 2 on the x-axis, the graph shows considerable difference to distinguish persuadables in both of them by a factor of 5% and more.

Result- Both 1 and 2 in V7 are ideal values to distinguish persuadables from the rest of the 3 categories.

Conclusion:

We have successfully conducted our project of predicting the customer's conversion rate on the starbucks dataset. We have applied the XGBoost algorithm on our dataset. After doing so, we determined the best features that are required to determine the design of a marketing campaign.
We found out that in V1, 1 would be the most ideal value to distinguish persuadables from the other 3 categories.30 in V2 is the most ideal value to distinguish persuadables from the rest of the 3 consumers.1 in V3 is the most ideal value to distinguish persuadables from the rest of the 3 categories.2 in V4 is the most ideal value to distinguish persuadables from the rest of the 3 categories.2 in V5 is the most ideal value to distinguish persuadables from the rest of the 3 categories.All the values on the x axis, 1, 2, 3 and 4 in V6 are ideal values to distinguish persuadables from the rest of the 3 categories.Both 1 and 2 in V7 are ideal values to distinguish persuadables from the rest of the 3 categories.

Our final Incremental Response Rate(IRR) value is 2.36% on the test set which is 0.48% more than the starbuck's baseline model. This means that 2.36% more customers purchased the product with the promotion, as compared to if they didn't receive the promotion.