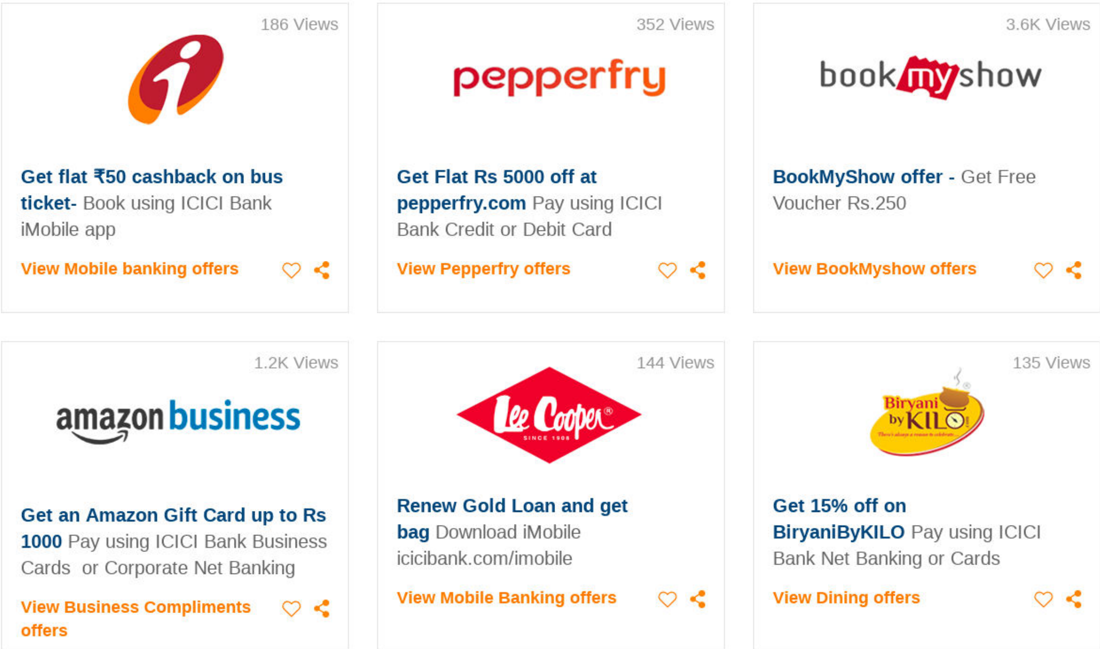
Here I will employ machine learning to design a promotional strategy for ICICI bank.

Can we increase ICICI bank profits by adopting a more selective promotional strategy?

Here is the situation: once every few days, ICICI bank will send out promotions to customers on the mobile app. These can be discount offers,cashback offers, or informational offers/vochers(Eg like thomas cook travel voucher pay).



There is often a cost associated with the promotion. In our case, a cashback or a discount promotion can result in "lost profits" for the firm, since the firm is sacrificing some profits to give the customers monetary incentives (free product or discount).

If the customers are already willing to purchase products from ICICI bank without being given any promotions, then we should abstain from giving this group of customers any promotions.
Ideally, we will want to send the promotions to individuals who are inclined to make purchases only when they are presented with offers, and we also want to avoid sending them to "sleeping dogs"."Sleeping dogs" are individuals who buy your product, but will stop doing so if they are included in our marketing campaign.
Our objective will be to formulate a promotional strategy that can identify individuals whom we should send promotions with the aim of maximizing profits for the firm.

To help us tackle this problem, we will be utilizing uplift models. Uplift models allow us to model the incremental impact of a treatment (this will be the promotion in our case) on a customer's purchase behavior.

"**uplift modelling**, also known as **incremental modelling** is a predictive modelling technique that directly models the incremental impact of a treatment (such as a direct marketing action) on an individual's behavior"

uplift modeling can help us identify individuals who will purchase our products only as a result of receiving a discount coupon or a personalized advertisement. Utilizing these models can help our firm maximize profits by keeping advertising cost to the minimum.These models offer us the best chance of improving the promotions effectiveness

Statistics for transaction behaviors would be aggregated on a individual basis. For each individual, I keep track of :

1.number of offers received

2.number of offers successfully completed

3.number of offers that were tried but not completed

4.percentage of offers successfully completed

5.percentage of offers tried

6.total spending for offer

7.total number of transactions made for offers

8.average spending per transaction for offers

These numbers were aggregated on a cumulative basis (all offers plus no offer), each offer type (id 0–9 plus id 10 representing no offer) and each offer family type (cashback, discount, voucher, ...).

Additional ratios were also computed, such as:

1.ratio of spending for offer type and offer family type over total spending

2.ratio of number of transactions for offer type and offer family type over total number of transactions

3.ratio of number of offer type and offer family type received over total number of offers received

In order to differentiate between the situation of

1) receiving an offer and not responding to it and

2) not receiving an offer at all, the number of offers received by each individuals were tracked.

Models

3 separate models need to be created, one model for each of the missing attributes: age, income and gender. The portion of the profile dataset without missing values would be used to train the models. Both the age and income models are regression problems, while the gender model is a multi-class classification problem. K-Fold cross validation with 5 folds can be used in the grid search process to optimize the models.

XGBRegressor and XGBClassifier are the models we can chose for the regression and classification tasks . These are tree-based non-linear models that are relatively fast and accurate.

The only major drawback to these models is that they do not innately extract feature interactions.

For example, if our datasets only track total spending for each offer type , but not for each family type , the XGBoost model will make modelling decisions based only on spending for each offer type. Hence, we will need to manually engineer these feature interactions if we want our models to take advantage of them.

The primary task will be to identify individuals who are likely to spend more money when receiving offers as compared to not receiving offers. Hence, we need a dataset that reflects how users respond in both promotional and non-promotional situations.We are also interested if changes in customers' behaviors can happen over time.Hence, I have chosen to aggregate customers' response on a monthly basis.

Every month, the dataset should track:

1.How much customers spent during offers' validity if they received offers. This figure could be 0 if they did not spend any money.

2.How much customers spent during periods of time when there were no offer. This figure could also be 0 if they did not spend any money.

Define Successful/Tried/Failed Offers

Before we can track how much money customers spent during the validity of promotions, we need to classify the offers according to their possible outcomes. There are 3 possibilities: successful, tried and failed.For an offer to be classified as successful, it has to be received, viewed and completed before the offer expired. This meant that the customer was aware of a promotion and was making transactions as a result.If the customer completed the offer before viewing it, the offer would not be classified as successful, since the customer was not influenced by the offer when making transactions.In the event that a customer made some transactions before viewing the offer, but did not spend enough to complete the offer. If he/she viewed the offer while it was still valid, and spent more money to complete it before it expired, the offer would be classified as successful as well.

Hence, the flow of events for a successful offer is:

- offer received -> optional: transactions made -> offer viewed -> transactions made -> offer completed -> offer expired

A tried offer is one that a customer viewed, spent some money before the offer expired but did not complete it. Hence the customer did not spend sufficient money to complete the offer's requirement.Since informational offer do not have an offer completion event, they will be treated as a tried offer if the customer viewed the offer and spend some money during its validity.

The flow of events for a tried offer is:

- offer received -> optional: transactions made -> offer viewed -> transactions made -> offer expired -> optional: offer completed

Failed offers will be offers that did not fall into the two previously mentioned categories.

For example, if an offer was received and viewed but no transactions were made before the expiry of the offer, the offer would be a failed offer.

If an offer was received but not viewed before its expiry, it would also be classified as a failed offer, even if money was spent during the offer's validity. This is because the customer is spending money without any influence from the offer.

Tracking the amount of money spent during promotions will be equivalent to finding the amount of money spent for successful and tried offers.

We can eliminate a great deal of false offers by keeping those that meet the following conditions:

1.time offer completed > time offer viewed > time offer received

2.(time offer viewed > time offer received) and (time offer completed is null)

3.both time offer viewed and time offer completed are null

False offers still exist at this stage, and further processing is needed.We can calculate the expiry time for all offers by adding the duration of offers to the receival times of offers.

Next, we can classify these offers into their probable outcomes: successful, tried or failed/false. Note that the classifications at this stage do not necessarily mean that the offers are truly successful or tried. We will need the transaction information later to find out.

Offers that meet the following condition will be classified as offers that are probably successful:

1.(time offer received ≤ time offer viewed) and (time offer viewed ≤ time offer completed) and (time offer completed ≤ time offer expiry)

Offers that meet the following conditions are classified as offers that are probably tried:

1.(time offer received ≤ time offer viewed) and (time offer viewed ≤ time offer expiry) and (time offer expiry < time offer completed)

2.(time offer received ≤ time offer viewed) and (time offer viewed ≤ time offer expiry) and time offer completed is null

The rest of the offers that did not meet these conditions are either failed or false offers and will be discarded.

Any offers with duplicated values for 'time_received', 'per_id' and 'offer_id' will be dropped with the exception of the first occurrence. It is unlikely that a person will receive the same kind of offer more than once at a time, and these duplicated entries are erroneous entries generated from the merging process. *monthly_transactions* already track how much customers spent during non-promotional times. Our goal here is to find out which months did customers not spend any money during non-promotional times.First, we generate all possible combinations of month number, person id and offer id.
We can then merge *monthly_transactions* data frame to *non_offer_trans* to find out which month-individual combinations that had no monetary transactions during non-promotional periods.Thus, we will obtain a monthly account of when customers did not spend money during non-promotional situations.

Aggregating them together

Finally, we can generate *monthly_data* by concatenating *monthly_transactions*, *monthly_failed_offers* and *no_offer_no_trans* together. The resulting dataset tracks on a monthly basis, how much each individual spent on the different promotions sent to them , as well as how much non-promotional spending they made.

Compute Profits and Generate Labels

Next, we have to compute the amount of profits generated for each instance of the dataset. We will first need to compute the number of offers each individual was exposed to every month. This allows us to compute the cost associated with the offers.

An easy way to do so is to check if individuals were exposed to more than 1 offer of each type in a month. By inspecting *transcript_received*, we note the following:

1.No individuals received the same offer type more than once in the same month.

2.If an individual received an offer that expires during the next month, he/she would not received an offer of similar type during the next month. For example, if an individual received ‘offer id 2’ at month 16 and the offer expires during month 17. He/she would not receive another ‘offer id 2’ at month 17.

Hence, we can conclude that customers were only exposed to a maximum of 1 occurrence of an offer type every month.This means that the cost in *monthly_data* is simply the reward of the promotion if it was completed.

We can calculate the amount of profits each individual generated for each offer type each month by following the 3 rules:

1.If the offer was successful, the profit would be the monthly revenue minus the cost of the offer. Note that informational offers have no cost.

2.If the offer was not successful, the profit would be the revenue generated in that instance.

3.If the transactions were not made as part of an offer, the profit would be the revenue since there are no cost involved.

The uplift model that we will be using, involves modelling the probabilities of profits for a given person and month in two situations:

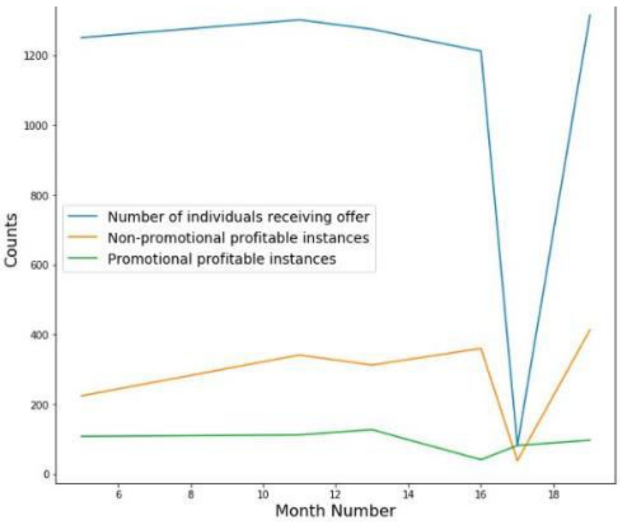
1.If the person receives an offer.

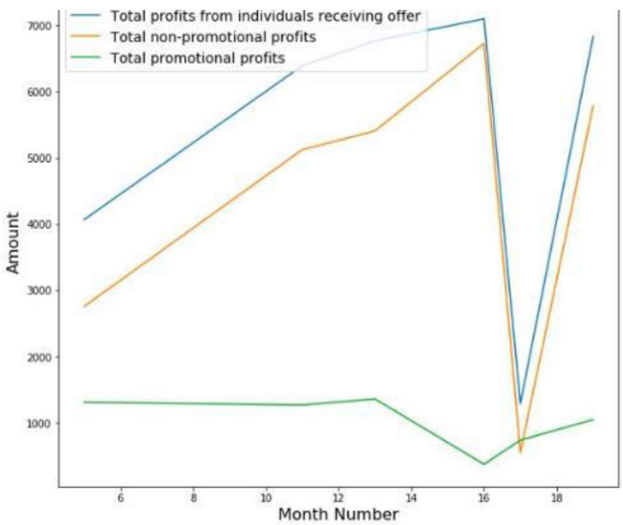
2.If the person did not receive an offer.

Because we want to predict the probability of profits, our labels (*has_profit)*will simply be an indicator variable indicating if there was a profit for that instance.

Exploratory Data Analysis

Here we have to plot different graphs which includes number of months agaist counts, amount.This graph should consists of number of indivisuals receiving offer , non-promotional profitable instances and promotional profitable instances.Like the figure shown below, we have to plot graph for different offers.





Feature Engineering part:

Customers often went through long periods of time without receiving promotions. Hence cumulative values and moving averages would be used to capture past transactional behaviors of customers.

Cumulative sums would be calculated for the following statistics:

1.total spending

2.number of transactions

3.profits

For example, the cumulative profits at time N will be

Cumulative Profit at Time N = Cumulative Profit at Time 0 + … + Cumulative Profit at Time N-1

Likewise the moving averages (rolling means) of the same statistics were calculated. For example:

Moving Average of Profit at Time N = Cumulative Profit at Time N / Number of Months

The cumulative statistics and moving averages would be computed for each ‘offer id’ including non-promotional situations (represented by ‘offer id 10’), as well as on an accumulative basis (all offers and no offer).

For example, we will compute:

- Cumulative Profit at Time N for Offer id 0
- Cumulative Profit at Time N for Offer id 1
- …
- Cumulative Profit at Time N for Offer id 10
- Total Cumulative Profit at Time N for All Offers id 0–10

In addition, the cumulative spending per transaction (total spending / total number of transactions) and cumulative profit per transaction (total profit / total number of transactions) will be added as well.

Any missing values will be filled with 0, since null values indicate that the customer had not yet made any transactions.

Indicator Uplift Model And Promotional Strategy

We will be using a single model to predict the probabilities of profits from both promotional and non-promotional exposure. During the training phase, an indicator variable is created to track if a data point from *monthly_data* belongs to a promotion or not.Each type of offer will have its own model, so a single indicator variable for each model will be sufficient.
Once the model is trained, it can be used to formulate our promotional strategy.

To predict whether an individual should receive a promotion when testing our strategy, we can predict the individual's profit probability when given the promotion by setting the indicator variable to 1. Next, we can predict the individual's profit probability when he/she is not given the promotion by setting the indicator variable to 0.The same model is used to predict the probability of profits during promotional and non-promotional periods. Only the inputs, specifically the indicator variable, are changed during the procedure.

If the difference in probability (also known as the uplift effect) is larger than 0, we will send the promotion. This is because the individual is more likely to generate profits when given a promotion as opposed to no promotions.

Uplift Effect = Probability of Profit When Given a Promotion—Probability of Profit When Not Given a Promotion

Alternatively, regression models can be used to model the expected amount of profits in promotional events versus non-promotional events. This can potentially tell us how much more profit we can expect to gain by sending an offer to an individual.

In addition, there are other types of uplift models that can be implemented for this task.

One such example will be to use two separate models to measure the uplift effect. In this scenario, one model will be trained on the promotional data while the other model will be trained on the non-promotional data. The difference between the predicted probabilities of the two models will indicate the uplift effect.

Metric

The performance of our promotion strategy will be determined using the Net Incremental Revenue (NIR), where:

NIR = Promotional Revenue—Cost of Promotion—Non-Promotional Revenue

which can also be expressed as

NIR = Promotional Profit—Non Promotional Profit

The NIR will be calculated based on individuals who should receive the offer according to our strategy. In other words, these are individuals with positive uplift values.

Thus, the NIR measures how much is made (or lost) by sending out the promotion to these individuals.

For example, let us assume that we are calculating the NIR for month 19. Suppose that our promotional strategy predicted customers with id 15 and 5550 will have positive uplift values and they should receive the promotion, and the actual transaction record for these individuals during month 19 is as follows:

	month_num	per_id	offer_id	monthly_amt_spent	cost	profit
4	19.0	15.0	0.0	0.00	0.0	0.00
5	19.0	15.0	10.0	8.69	0.0	8.69
824	19.0	5550.0	0.0	28.20	5.0	23.20
825	19.0	5550.0	10.0	16.76	0.0	16.76

The NIR will be calculated as such:

NIR = ($0+$23.20)—($8.69+$16.76) = -$2.25

Results

We will now compare the results obtained from the baseline strategies and our uplift models.

The baseline strategy will be the original strategy employed during the study. In other words, everyone who received the offer during the actual experiment will receive the offer in the baseline strategy.Our model's goal would be identify a smaller subset of these individuals who were likely to spend more when given a promotion as opposed to when they were not given a promotion. In other words, the uplift model will send the promotions only to individuals with positive uplift values.Ideally, ICICI bank can maximize its profits by restricting the promotions only to the most promising customers.

Discount 10/20/5 (Offer ID 0)

Offer ID 0 is a discount promotion with a difficulty of $20, a reward of $5, and a validity period of 10 days.

Baseline Strategy ~ Validation NIR: $108.70, Test NIR: -$4,889.48

Uplift Model ~ Validation NIR: $72.83, Test NIR: -$2,163.47

In above cases, we were able to make significant improvements over the baseline strategies test months NIRs.

Implementing above for different promotion types, we were able to find strategies that were profitable during the validation and test months.