

Project Report

Coupon Redemption Prediction

Monil Gudhka

8th October, 2019

Contest:

<https://datahack.analyticsvidhya.com/contest/amexpert-2019-machine-learning-hackathon/>

Repository: https://github.com/monilgudhka/coupon_redemption_prediction



[Problem Statement](#)

[Dataset](#)

[Data Cleaning](#)

[Missing Data](#)

[Outliers](#)

[Data Merging](#)

[Coupon Information](#)

[Customer Behaviour](#)

[Campaign and Customer Information](#)

[Campaign and Coupon specific Customer Behaviour](#)

[Deriving Features](#)

[Exploratory Data Analysis](#)

[Initial Questions](#)

[Redemptions](#)

[Campaigns](#)

[Customer's Information](#)

[Customer Transactions](#)

[Modelling](#)

[Missing Customer's information](#)

[Customer behaviour](#)

[Customer behaviour in Percentage](#)

[Hyper-parameter Tuning](#)

[Ensemble](#)

[Conclusion](#)



Problem Statement

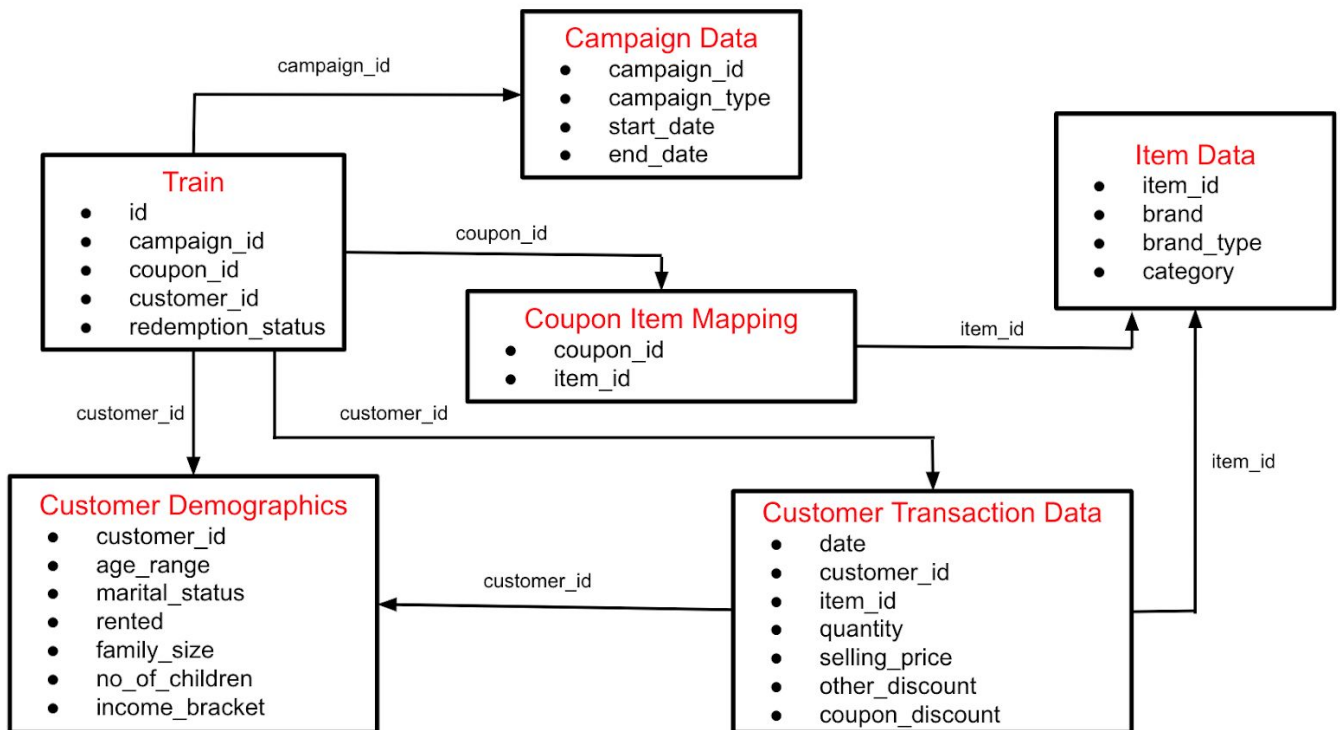
Discount marketing and coupon usage are very widely used promotional techniques to attract new customers and to retain & reinforce loyalty of existing customers. The measurement of a consumer's propensity towards coupon usage and the prediction of the redemption behaviour are crucial parameters in assessing the effectiveness of a marketing campaign.

The client's promotions are shared across various channels including email, notifications, etc. A number of these campaigns include coupon discounts that are offered for a specific product/range of products. The retailer would like the ability to predict whether customers redeem the coupons received across channels, which will enable the retailer's marketing team to accurately design coupon construct, and develop a more precise and targeted marketing strategies.

The evaluation metric for this competition is an area under the ROC curve between the predicted probability and the observed target across all entries in the test set. Since, we do not have access to the output of test set. Hence, Evaluation will be done by submitting the solution in the Contest.

Dataset

Here is the schema for the different data tables available. The detailed data dictionary is provided next.



The client has provided the following information, the task is to predict whether customers will redeem the coupon for the campaigns with id 18-25:

1. **train.csv**: Contains the coupons offered to the given customers under the 18 campaigns, test.csv contains all the following features except the target variable

Variable	Definition
id	Unique id for coupon customer impression
campaign_id	Unique id for a discount campaign
coupon_id	Unique id for a discount coupon
customer_id	Unique id for a customer
redemption_status	(target) (0 - Coupon not redeemed, 1 - Coupon redeemed)

2. **campaign_data.csv**: Campaign information for each of the 28 campaigns

Variable	Definition
campaign_id	Unique id for a discount campaign
campaign_type	Anonymised Campaign Type (X/Y)
start_date	Campaign Start Date
end_date	Campaign End Date

3. **coupon_item_mapping.csv**: Mapping of coupon and items valid for discount under that coupon

Variable	Definition
coupon_id	Unique id for a discount coupon (no order)
item_id	Unique id for items for which given coupon is valid (no order)

4. **customer_demographics.csv**: Customer demographic information for some customers

Variable	Definition
customer_id	Unique id for a customer
age_range	Age range of customer family in years
marital_status	Married/Single
rented	0 - not rented accommodation, 1 - rented accommodation
family_size	Number of family members
no_of_children	Number of children in the family
income_bracket	Label Encoded Income Bracket (Higher income corresponds to higher number)

5. **customer_transaction_data.csv**: Transaction data for all customers for duration of campaigns in the train data

Variable	Definition
date	Date of Transaction
customer_id	Unique id for a customer
item_id	Unique id for item
quantity	Quantity of item bought
selling_price	Sales value of the transaction
other_discount	Discount from other sources such as manufacturer coupon/loyalty card
coupon_discount	Discount availed from retailer coupon

6. **item_data.csv:** Item information for each item sold by the retailer

Variable	Definition
item_id	Unique id for item
brand	Unique id for item brand
brand_type	Brand Type (local/Established)
category	Item Category

Data Cleaning

After analysing the dataset, below issues were found.

Missing Data

For some customers, "no_of_children" and "marital_status" fields are missing. For "no_of_children", making an assumption that if it is NaN then it means that customers have no/zero children. Hence, filling missing "no_of_children" with zero.

Marital status of the customers were calculated using family size and no of children. If a customer has a member who is not his/her child than consider that extra member to be his/her spouse. Hence, If there is an extra member in the customers family then that customer is married.

Train and Test set contains many customers whose information is not available. Keeping variables to be NaN for the prediction algorithm to handle them.

Outliers

Data contains many outliers in the customer's transactions. Since, there are many outliers, Prediction algorithm needs to be trained with outliers.

Data Merging

Coupon Information

Since, Merging coupon information with training and test data is many-to-many mapping, it needs to be reduced to one-to-many mapping. Extracting below summary variables from coupon information.

c_unique_items, c_unique_brand, c_freq_brand, c_rare_brand, c_items_freq_brand, c_items_rare_brand, c_unique_brandt, c_freq_brandt, c_rare_brandt, c_items_freq_brandt, c_items_rare_brandt, c_unique_category, c_freq_category, c_rare_category, c_items_freq_category, c_items_rare_category, c_coverage_item, c_coverage_brand, c_coverage_brandt, c_coverage_category

Customer Behaviour

Similar to coupon information, extracting below summary variables from customer transactions considering them as the customers buying behaviour.

overall_unique_items, overall_items, overall_quantity, overall_sprice, overall_bprice, overall_odiscount, overall_cdiscount, overall_tdiscount, overall_sprice_pq, overall_bprice_pq, overall_odiscount_pq, overall_cdiscount_pq, overall_tdiscount_pq, overall_unique_brand, overall_freq_brand, overall_rare_brand, overall_items_freq_brand, overall_items_rare_brand, overall_unique_brandt, overall_freq_brandt, overall_rare_brandt, overall_items_freq_brandt, overall_items_rare_brandt, overall_unique_category, overall_freq_category, overall_rare_category, overall_items_freq_category, overall_items_rare_category, overall_coverage_item, overall_coverage_brand, overall_coverage_brandt, overall_coverage_category, overall_podiscount, overall_pcdiscount, overall_ptdiscount, overall_podiscount_pq, overall_pcdiscount_pq, overall_ptdiscount_pq

Campaign and Customer Information

Campaign information and customer information both have one-to-many mapping with train and test data. Hence, they can be merged easily using left join.

Campaign and Coupon specific Customer Behaviour

Up till now, all the features were related to customer, campaign and coupon in general. Features related to combination of all three are required which can be extracted from customer's transaction, so that model can predict the behaviour of customer for a campaign.

Below table represents the analysis of the date ranges in the dataset.

	Customer transaction	Campaign Start	Campaign End
Start date	2012-01-02	2012-08-12	2012-09-21
End date	2013-07-03	2013-10-21	2013-12-20

If we consider the difference in start date of customer transaction and campaign starts, its 223 days. And the difference in end date of the same is 110 days. Hence, we can consider the

customer's transaction between 223 days to 110 days ago for a given campaign to extract the features.

range_unique_items, range_items, range_quantity, range_sprice, range_bprice, range_odiscount, range_cdiscount, range_tdiscount, range_sprice_pq, range_bprice_pq, range_odiscount_pq, range_cdiscount_pq, range_tdiscount_pq, range_unique_brand, range_freq_brand, range_rare_brand, range_items_freq_brand, range_items_rare_brand, range_unique_brandt, range_freq_brandt, range_rare_brandt, range_items_freq_brandt, range_items_rare_brandt, range_unique_category, range_freq_category, range_rare_category, range_items_freq_category, range_items_rare_category, range_coverage_item, range_coverage_brand, range_coverage_brandt, range_coverage_category, range_podiscount, range_pcdiscount, range_ptdiscount, range_podiscount_pq, range_pcdiscount_pq, range_ptdiscount_pq, redemption_ratio


We can also filter transactions within above range with items that are covered within a given coupon.

range_coupon_unique_items, range_coupon_items, range_coupon_quantity, range_coupon_sprice, range_coupon_bprice, range_coupon_odiscount, range_coupon_cdiscount, range_coupon_tdiscount, range_coupon_sprice_pq, range_coupon_bprice_pq, range_coupon_odiscount_pq, range_coupon_cdiscount_pq, range_coupon_tdiscount_pq, range_coupon_unique_brand, range_coupon_freq_brand, range_coupon_rare_brand, range_coupon_items_freq_brand, range_coupon_items_rare_brand, range_coupon_unique_brandt, range_coupon_freq_brandt, range_coupon_rare_brandt, range_coupon_items_freq_brandt, range_coupon_items_rare_brandt, range_coupon_unique_category, range_coupon_freq_category, range_coupon_rare_category, range_coupon_items_freq_category, range_coupon_items_rare_category, range_coupon_coverage_item, range_coupon_coverage_brand, range_coupon_coverage_brandt, range_coupon_coverage_category, range_coupon_podiscount, range_coupon_pcdiscount, range_coupon_ptdiscount, range_coupon_podiscount_pq, range_coupon_pcdiscount_pq, range_coupon_ptdiscount_pq

Deriving Features

Deriving more features which represents the change in customer behaviour.

diff_range_unique_items, diff_range_items, diff_range_quantity, diff_range_sprice, diff_range_bprice, diff_range_odiscount, diff_range_cdiscount, diff_range_tdiscount,



*diff_range_podiscount, diff_range_pcdiscount, diff_range_ptdiscount, diff_range_sprice_pq,
diff_range_bprice_pq, diff_range_odiscount_pq, diff_range_cdiscount_pq,
diff_range_tdiscount_pq, diff_range_podiscount_pq, diff_range_pcdiscount_pq,
diff_range_ptdiscount_pq, diff_range_unique_brand, diff_range_unique_brandt,
diff_range_unique_category, diff_range_coverage_brand, diff_range_coverage_category*

Features that represents the match of customer behaviour with that of coupon.

*match_freq_brand, match_rare_brand, match_freq_brandt, match_rare_brandt,
match_freq_category, match_rare_category*

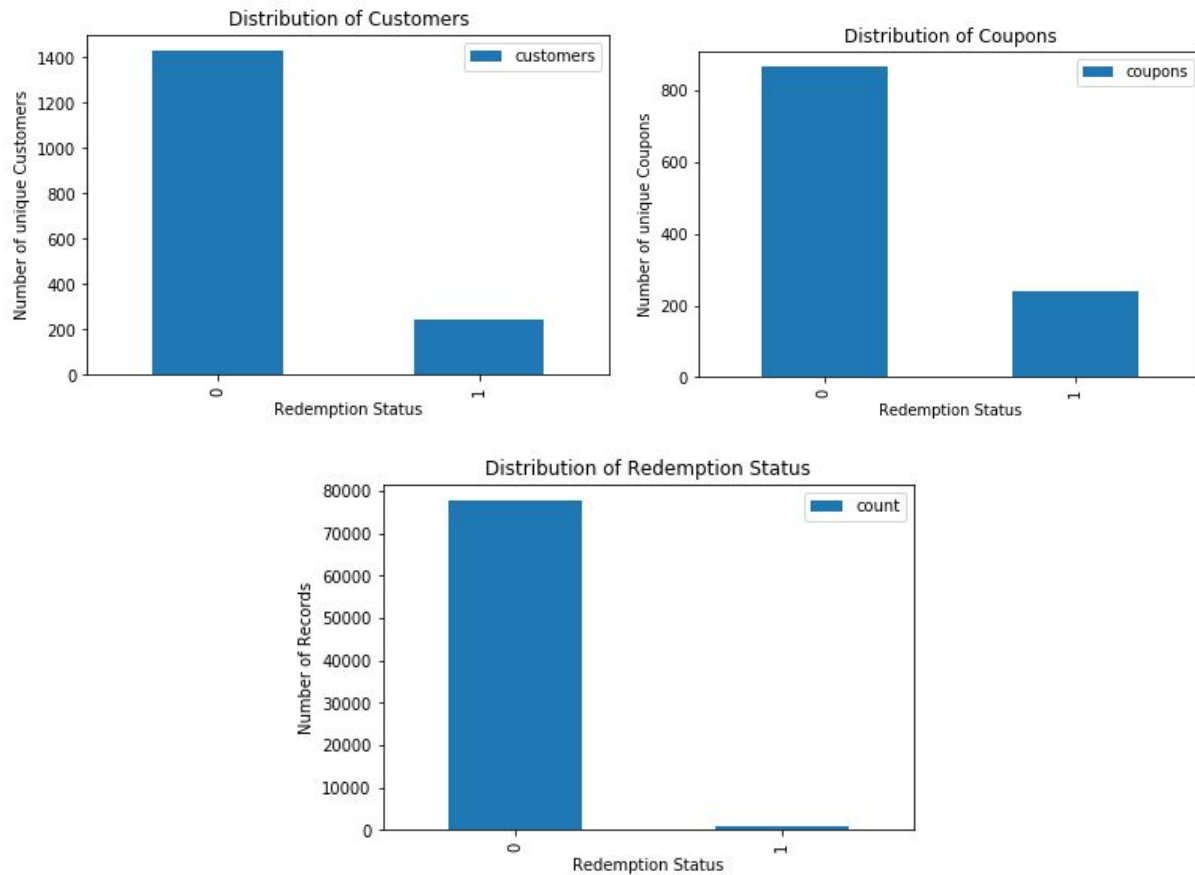


Exploratory Data Analysis

Initial Questions

1. Distribution of redemption status with respect to coupon, customer and campaign?
2. What is the trend in the campaign duration?
3. How many customers whose information is not present?
4. What is the trend in the Customer transactions?

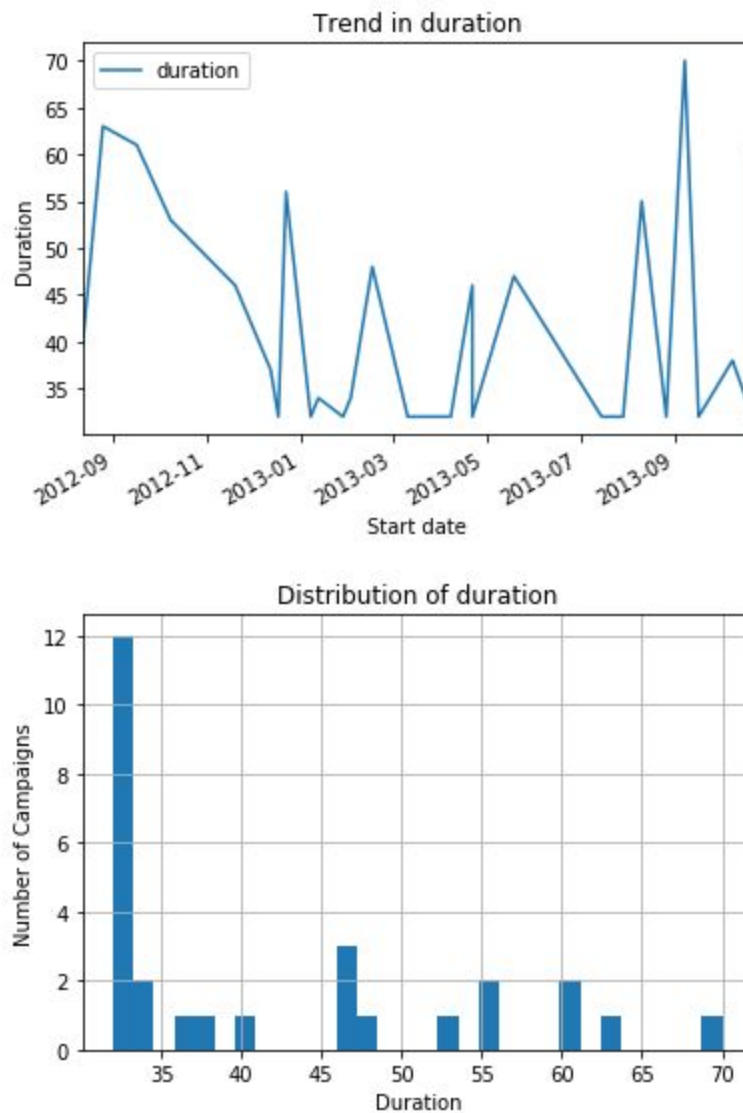
Redemptions



Above plots represents the distribution of Customers, Coupons and Campaigns with respect to redemption status. Below are findings from above plots:

1. There are 627 (78%) coupons which are not redeemed by any customers
2. There are 1181 (85%) customers who has not redeemed any coupons
3. The data is highly Imbalanced

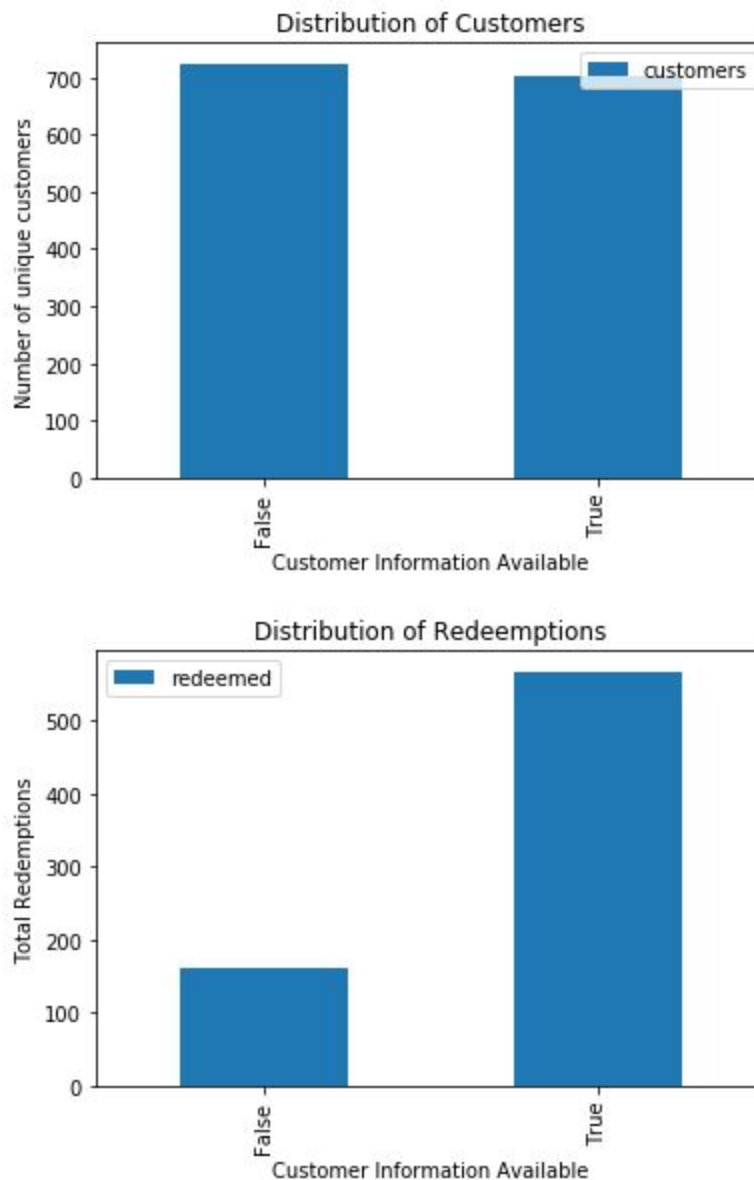
Campaigns



Above plots represents the Trend and distribution of Campaign's duration respectively. Below are findings from above plots:

1. The duration of Campaigns in the initial days were longer
2. Later on, Campaigns with both short and long durations were started
3. Most of the campaigns fall in duration of 35 and less days
4. There is one Campaign with duration of 70 days

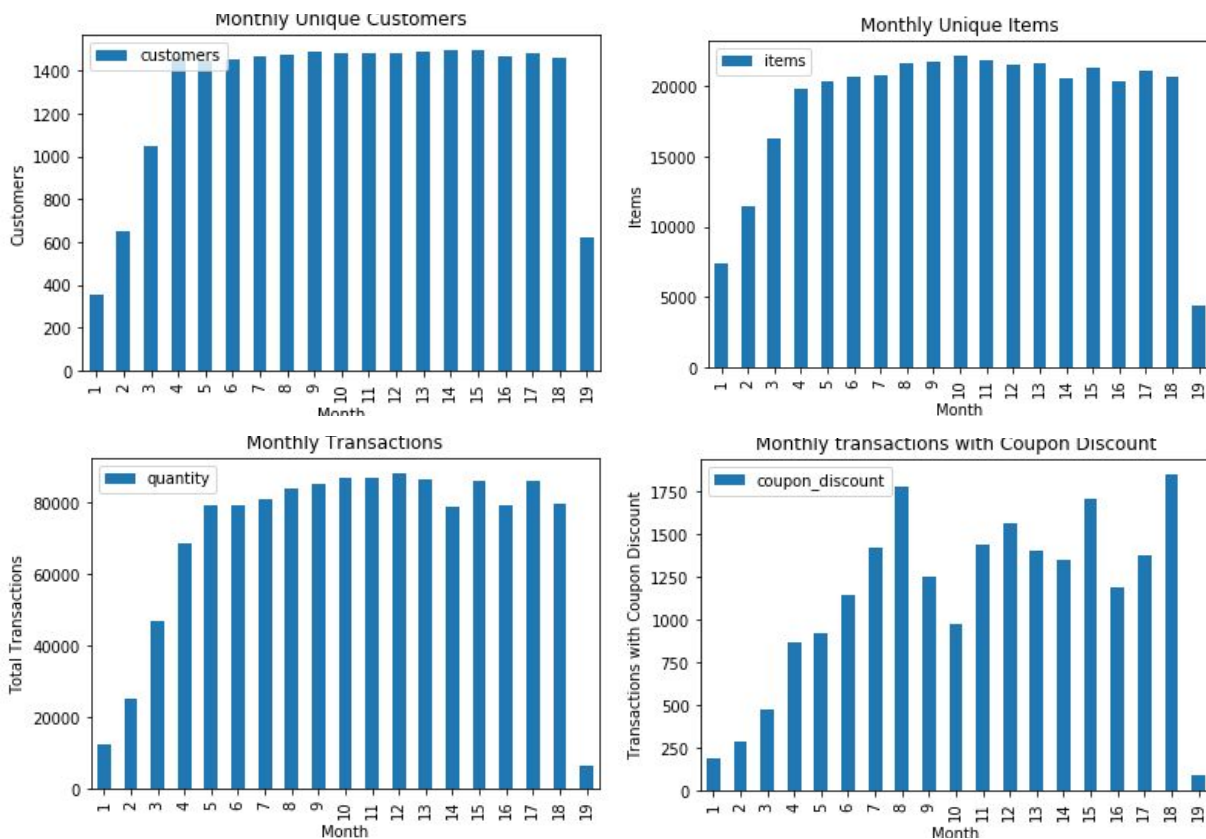
Customer's Information



First plot represents the distribution of unique customers and total redemptions made by them with respect to the information availability respectively. Below are findings from both the plots:

1. Almost 50% of customer's information is not available
2. Customer's whose information is available tends to redeem more coupons than customer's with no information

Customer Transactions



Above plots represents in the monthly trends in transactions with respect to customers, items, total transactions, total transactions with coupon discount respectively. Below are findings from above plots:

1. There are not much transaction in the initial 3 months, may be client has just started its business
2. Month 19 can't be considered in any analysis because it only contains 3 days transactions
3. Customers, Items and total transactions grew in initial 3 months and then remains almost stable
4. Transactions with coupon discount has fluctuations

Modelling

The Client wants to predict whether customers redeem the coupons received across channels. This is a binary classification problem where model needs to predict the redemption status as a 0 or 1 on input campaign, customer and coupon combination.

Since, the feature set contains many categorical variables. Hence, algorithm, like lightGBM, which can handle categorical values without any encoding strategy is useful. Many experiments targeting specific set of features were performed.

Missing Customer's information

One experiment was to check the performance of model with and without the missing values. It was found that model with replacing missing values with default values performed better on test set. Although removing features performs good on validation set but not on test set.

Model	Validation Set	Test Set
Removing Features with Missing values	0.9975	0.8739
Keeping Missing values as NaN	0.9974	0.8973
Replacing Missing values with default values	0.9970	0.9016

Customer behaviour

Since, Customer behaviour in a summary of all transactions done by customers. Hence, experiment to check the performance of model with and without customer behaviour. It was found that model with customer behaviour performed better than model without customer behaviour on test set

Models	Validation Set	Test Set
With Customer Behaviour	0.9970	0.9016
Without Customer Behaviour	0.8647	0.8690

Customer behaviour in Percentage

One experiment was to check customer behaviour in terms of percentage only. It was found that model with both percentage and count features performed better on test set

Models	Validation Set	Test Set
With Percentage and Count	0.9970	0.9016
Only Percentage	0.9952	0.8803

Hyper-parameter Tuning

Uptil now, models were using default parameters. After, tuning the parameters of lightGBM algorithm. It was found that below parameters performed better.

Parameter	Validation Set	Test Set
default	0.9970	0.9016
learning_rate: 0.08 num_leaves: 25 max_depth: 7 pos_bagging_fraction: 1.0 neg_bagging_fraction: 0.09 bagging_freq: 50 feature_fraction: 0.4 early_stopping_round: 100 num_iteration: 3000	0.9970	0.9060

Ensemble

Average of outcomes from lightGBM models with num_leaves 15, 20 and 25 was also evaluated on test set. It gave the result of **0.9132**.



Conclusion

After performing different experiments with features, missing values, parameters, etc. The ensemble of tuned lightGBM with num_leaves 15, 20 and 25 performs better than others. The evaluation metric of **area under the ROC curve** final model is getting the score of **0.9132**. The same model got private score of **0.8979**.

Further Improvements can be done to improve the model. Below are some ideas that can be explored.

- The model uses features, where each feature has a small contribution towards the model performance. Merging the features will result in better speed of model and hence, more experiments can be performed.
- Different way of handling many-to-many relations between coupons and customer's transactions.
- Hyper-parameter tuning might lead to local minima, further tuning can be tried for better results.
- Confusion matrix was not analysed. It can be analysed and fix the models for the issues.

