# Technical Report

Market Optimization with Cost Sensitive Learning

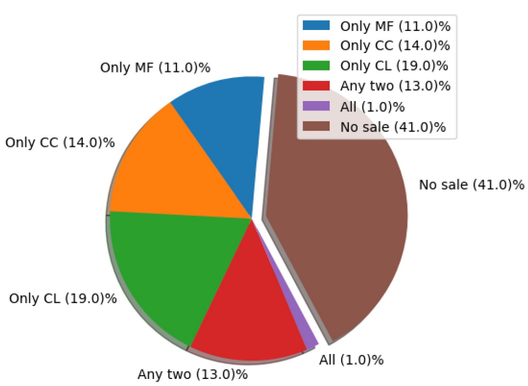
Bhanu Angam

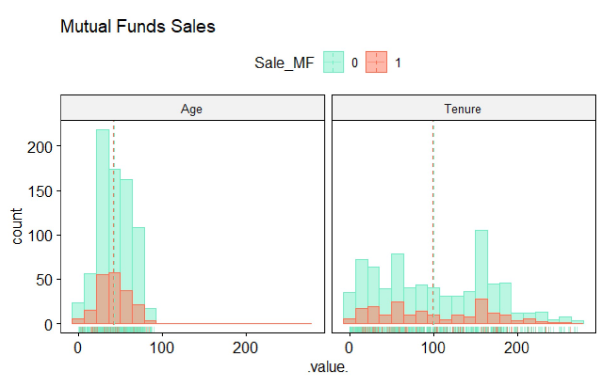
### Introduction

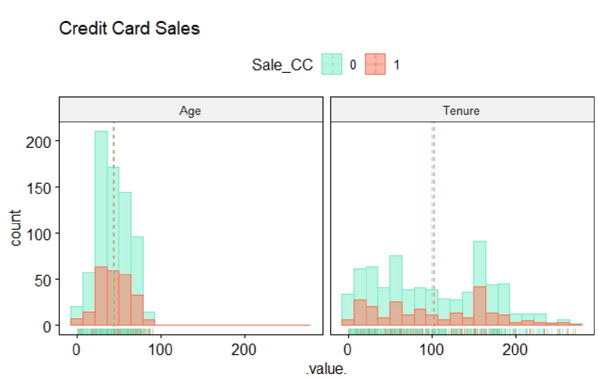
In this constructive analysis, we examine the development of propensity models for the optimization that decide on the details of direct -marketing campaign. Several models are compared based on advantages of certain type of metrics during this process. In this type of modelling the metric plays a vital role and also decides the revenue generated by the campaigns. It also allows the retention ability of the customers who doesn't show the potential qualities but actually potential candidate among all. During the process the type of sampling methods for balancing are also compared with the out put from the models. State of the art technique is applied for the expected revenue estimation from the propensities of the target audience.

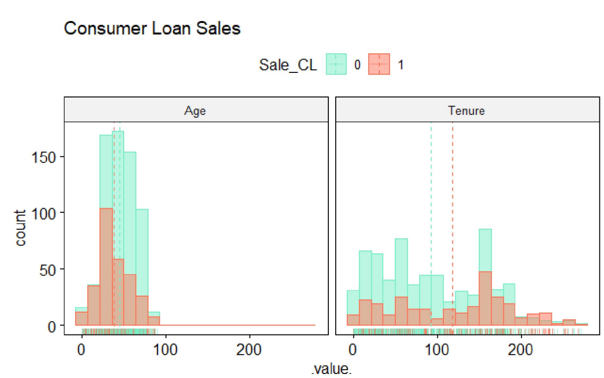
## Exploratory Analysis

The data consists of 4 tables with few Demographics, Account status, Transaction data and target data of 3 products and their respective revenues. The demographic data is not satisfying the logic. There are 34 clients with age less than the tenure and 8 clients with credit cards below age 10 years. All there obs are not touched because of many reasons. Separate detailed reports for each variable is provided in pdf format in the project directory.



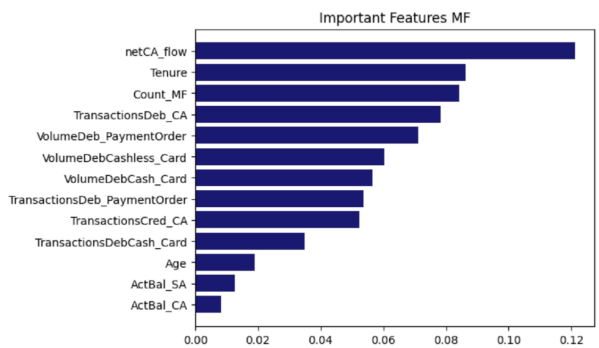



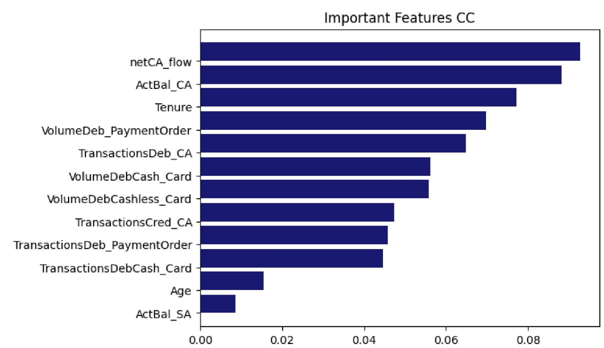
There are many feature in the Inflow_Outflow table that are inter correlated, because they are computed from the other existing feature. All such couples are separated by keeping only one among them. The Debited and Transaction couples are subtracted and generated new net-inflow and net-outflow features by removing the couples. There are 28 observations of clients missing from the transaction tables are completely discarded as they bring no information into the data. Correlation plots are monitored tried to keep the multi-col linearity minimum by transforming the features.

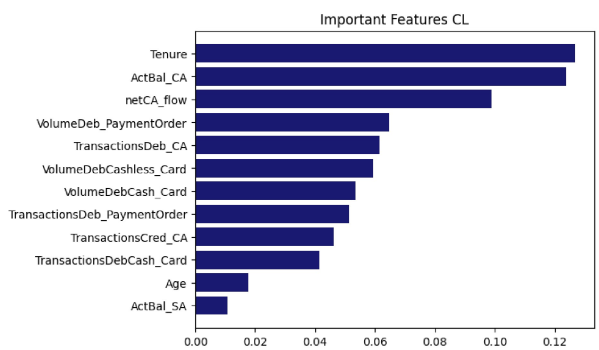
## Data Preperation

### Feature Engineering

The data sets are merged on the common column 'Client' and splitted into Train data set with available targets with and Target data set that have no targets. Feature selection is very important here to reduce the over fitting and multi col linearity, The available feature are not giving enough information are removed from the data sets. The train data is split into Train and test data sets using stratification and Random forest with grid search is used to optimize the fit. Important variables are selected based on threshold for each product. The selected features are used to subset and prepare 4 sets for 3 products and 1 without any product. Additionally PCA and LDA are performed to study the behavior of dimensional reduction, both the methods failed to find the maximum separability between the classes, results are attached in appendix.




### Over and Under Sampling

The train data is again split into two sets for training the models and testing them. The imbalance problem is tried to solve with various Over and under-sampling techniques. Individually Oversampling SMOTE and Under sampling Randomly are used to train the models, Combination of Over and under is also made and compared all with the base line models. The Sampling techniques didn't improve the models rather they gave inaccurate results in some cases. Therefore it is decided not to use sampling techniques and rely upon Cost sensitive learning discussed below.

## Propensity Modeling

The response rate for all the 3 products is quite low. Classical models fail to classify the data with accuracy. we need a robust technique to take this class imbalance into account. The other problem with Imbalanced data is that misconception of the metrics. If you focus on the over all accuracy the models looks like they are performing well but the confusion matrix shows up with high number of False negatives or false positives. As the Majority class is negative (0), the model tends to over fit and predict majority class most of the times this increases the overall accuracy, but the important true positives(minority class) is neglected. There are many other metrics that are useful in this situation which focus more on the true positive rate and less on the accuracy.

```
## Name: Sale_CC, dtype: int64
##     Clients:
##     Total: 951
##     Negatives: 714
##     Positive: 237 (24.92% of total)
## Name: Sale_MF, dtype: int64
##     Clients:
##     Total: 951
##     Negatives: 758
##     Positive: 193 (20.29% of total)
## Name: Sale_MF, dtype: int64
##   Clients:
##     Total: 951
##     Negatives: 662
##     Positive: 289 (30.39% of total)
```

Models should also be updated by introducing weights to the cost function.If higher weights are given to the minor class, the cost fiction at each epoch gives high weightage to the miss-classification of minor class which in turn adjusts the model weights with high values during back propagation process. The best weights are calculated on the basic thumb rules based on the response rate of the sample.

## Cost Sensitive Learning

Correct classifications have a cost of zero, that is, $c_{00} = c_{11} = 0$. Misclassification costs are however in practice difficult to estimate.Prediction of potential customers as non potential (false negative) involves a loss directly related to the amount of the Revenues, but also on further strategies and loss of resources, and on the company reputation. At the same time, identifying non potential customer as potential (false positive) causes inconvenience to customers, generates useless investigation costs, and also impacts the company reputation.

In cost-sensitive imbalanced problems, the most popular heuristic approach to estimate the costs lies in utilizing the imbalance ratio (IR). Let us denote by $\mathcal{X}$ the imbalanced dataset, with $\mathcal{X}_0$ and $\mathcal{X}_1$ being the subsets of samples belonging to the majority and minority class, respectively. The IR of the dataset $\mathcal{X}$ is defined as Guillaume, Fernando, Christos:

$$IR = \frac{|\mathcal{X}_1|}{|\mathcal{X}_0|}$$

where $| \cdot |$ denotes the cardinality of a set. In this setting, $C(i, j) = IR$ and $C(j, i) = 1$, where the minority class is the i-th class, and the majority class is the j-th class. It is worth noting that using the IR as the cost for the majority class balances the overall cost of the two classes, that is, $|\mathcal{X}_1| = IR * |\mathcal{X}_0|$. The resulting cost matrix for a 2-class problem is given below.

Using the IR to set the misclassification costs is usually a good heuristic. It however has some limitations, in particular related to small sample size, class overlapping, and noisy or borderline instances {cite}‘fernandez2018learning‘. A common complementary practice consists in considering the misclassification costs as a hyperparameter to be identified through model selection.

Python sklearn provides support for cost-sensitive learning for most baseline classifiers thanks to the ‘class_weight‘ parameter. The parameter allows to specify costs in three different ways:

- ‘None‘: The misclassification costs are set to 1 (default)
- ‘balanced‘: The costs are set according to the imbalance ratio (as in Fig. below)
- ‘{0:c10, 1:c01}‘: The misclassification costs are explicitly set for the two classes by means of a dictionary.

The use of class weights usually implies a modification in the loss function of the learning algorithm. The modification depends on the type of algorithm. By strongly penalizing mistakes on the minority class, cost-sensitive learning improves their importance during the classifier training step. This pushes the decision boundary away from these instances, allowing to improve generalization on the minority class fernandez2018learning,gupta2020class.

Simple thumb rule for weights calculation:

```
# Scaling by total/2 helps keep the loss to a similar magnitude.

# The sum of the weights of all examples stays the same.

weight_for_0 = (1 / neg) * (total / 2.0)

weight_for_1 = (1 / pos) * (total / 2.0)

class_weightcc = {0: weight_for_0, 1: weight_for_1/1.35}
```

### Choosing a right Metric

Three types of models weighted Logistic Regression, XGBoost and ANN are used with the class weights and tuned the hyper parameters with the *F1-beta* score and couple of metrics.

Since the response rate is lower than 25% the default metrics like AUC_ROC and accuracies will degrade the model performance in classifying the minor class. We need a metric that is a trade-off between the True positives and true Negatives. Something in between the Precision and Recall.

Effect of Precision:

Correctly predicting potential customers for a product by reducing the Falsely predicting that a customer is potential

- Precision will keep the non potential customers away from the target set for marketing campaign
- But there is a chance of loosing potential customers from the campaign list.

Effect of Recall:

- Comsiders maximum potential customers into prediction score without losing them and include them in the campaign list
- Bu there is a chance of including Non-potential customers into the campaign list.

Therefore taking some non potential customers is okay unless you will not neglect the potential customers. Recall can be given high weight by using *F1-beta* score.

The accuracies and scores for each model is given below for Logistic regression:

Table 1: Scores for Class Weighted Logistic Propensity Models

| | F1 | $F_{Beta2}$ | AUC/ROC | Class Weights |
|---|---|---|---|---|
| Mutual Fund | 0.43 | 0.46 | 0.59 | 0.35 : 0.64 |
| Credit Card | 0.39 | 0.38 | 0.62 | 0.23 : 0.76 |
| Consumer Loan | 0.44 | 0.47 | 0.54 | 0.31 : 0.68 |

| | F1 | $F_{Beta2}$ | AUC/ROC | Class Weights |
|---|---|---|---|---|
| No product | 0.60 | 0.74 | 0.61 | 0.30 : 0.69 |

Table 2: XGBoost models

| | F1 | $F_{Beta2}$ | AUC/ROC | Weights |
|---|---|---|---|---|
| Mutual Fund | 0.44 | 0.45 | 0.61 | 1 : 1.9 |
| Credit Card | 0.43 | 0.56 | 0.63 | 1 : 1.4 |
| Consumer Loan | 0.48 | 0.63 | 0.59 | 1 : 1.4 |
| No product | 0.59 | 0.75 | 0.62 | 1 : 1.3 |

Table 3: ANN

| | F1 | $F_{Beta2}$ | AUC/ROC | Class Weights |
|---|---|---|---|---|
| Mutual Fund | 0.658 | 0.66 | 0.65 | 0.63 : 2.11 |
| Credit Card | 0.67 | 0.67 | 0.64 | 0.78 : 1.71 |
| Consumer Loan | 0.62 | 0.63 | 0.60 | 0.61 : 0.97 |
| No product | 0.60 | 0.76 | 0.63 | 0.70 : 0.96 |

Comparing the results from the three set of models regarding the computation. XGboost and Logistic took a while to process where as ANN took less time to run, The specification and architecture of the networks are optimized based on the type of product by adjusting at every un with the help of loss performance graphs. Comparatively ANN gave good results for all the four cases and the models are efficient with balanced prediction between imbalanced classes. The trade off is well made using ANN models. Where as Logistic regression showed better AUC scores but in fact the sensitivity is quite low. This is the example for the general metric pitfall as we discussed above. Logistic did well with specificity but poor with sensitivity. XGBoost performed moderately better among three. It tried to balance the Sensitivity and Specificity well but not as much as ANN did.

Initial thoughts were ANN would over fit easily but thanks to the cost effective learning the class weights have really improved the loss for each epoch.

## Revenue

The best models from ANN for each product are used to predict the propensities of the target clients. The top clients for each product are saved. The continuous revenue of each product did not correlate wit any of the features much, An attempt is made to regress the revenues on the train data. The accuracies and adjusted $R^2$ is less than significance level and the estimates are biased with heavy variance. Therefore it is decided to choose different strategy called Heuristic Revenue optimization by Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. These heuristics are either variants of the algorithms used in practice for application in a bank or specifically developed based on the structure of the problem.

$$M = *maximize* \sum_{j=1}^{n} \left[ \left( P_j - C_j \right) u_j - f_j \right]$$

1. For each product j, compute the average revenue Pj and the average cost Cj
2. Solve the resulting integer-programming formulation (M)
3. Sort products such that for two products j and k, j < k if and only if Pjuj   Pkuk
4. For each product j, sort clients such that for two clients i and l, i < l if and only if pij   plj

5. consider the products following the order obtained in line 3; offer each product j to the first uj active clients in the sequence given by line 4
6. keep this solution if it is feasible and if its total profit is greater than 0; otherwise, output the trivial solution.

The Maximum revenue among all three products is chosen as best revenue for the client. and the type of offer to propose.

### Targeting Strategy

- Since there is condition to contact a customer only once, we strike with the offer which has maximum estimated revenue among adjusted revenues from the three products.
- We sort the clients based on the best maximum revenue and choose 15 pct out of all targets or 100 clients.

### Estimated Total Revenue

The top 100 clients with maximum calculated Revenue is used to estimate the total revenue from the Campaign.

***The estimated Total Revenue from the above propensity models and heuristic strategy is 783.92***

### Estimating Revenue from previous Campaign

- The Revenue from previous setting is estimated in the same way using the "Sales_Revenues" table data.
- Comparing the previous Estimated revenue with the obtained revenue from heuristic process.
- The current Estimated Revenues is 783.9 ~ 784
  - Revenue from previous setting(existing data) is 740
- Shows that they both match with each other.

## Conclusion

The aim of this study is to study and estimate the propensities of the customers who are not the targets before thereby providng the right marketing strategy fo those clients with higher propensities and achieve maximum revenue out of expenditures.

Over-sampling, Under-sampling and mixture of both (Combine) are used to get most important feature among useless features. But these techniques failed to show significant results in the simple models like Logistic regression and XGBoost. Even though ANN are prone to over fitting and difficult to work with large number of hyperparameters, gave best results compared to other models. Thanks to cost sensitive learning and class weighting technique that gave good results when used with ANN. Even though the final accuracies are low, the f1-scores and recall of the models are quite satisfactory to rely upon and plan marketing strategies.

The Final revenues are estimated based on the Heuristic process gave very good result when compared with the revenues from the existing sales data. The final best revenues and target offers are saved to the project folder as follows:

- 'client_offer_target.csv' : Total target Clients who whith high propensities to buy at least 1 product
- 'clients_propensity_CC.csv' : High potential top 100 target clients for the Credit Card product
- 'clients_propensity_CL.csv' : High potential top 100 target clients for the Consumer Loan product
- 'clients_propensity_MF.csv' : High potential top 100 target clients for the Mutual Funds product

***Final Expected Revenues based on this strategy is 783.92.***