

# **Empirical Evaluation of Sampling and Classifier Selection for Predictive Modeling for Default Risk**

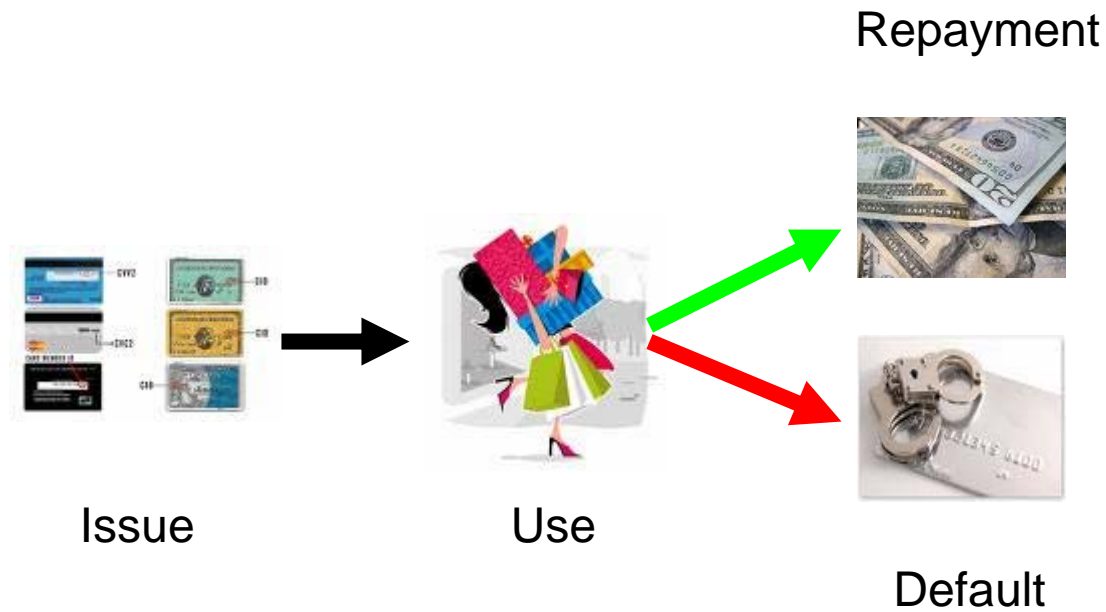
**Dr. Satchidananda Sogala, Ph.D**

**Head, Risk Solutions & Research**

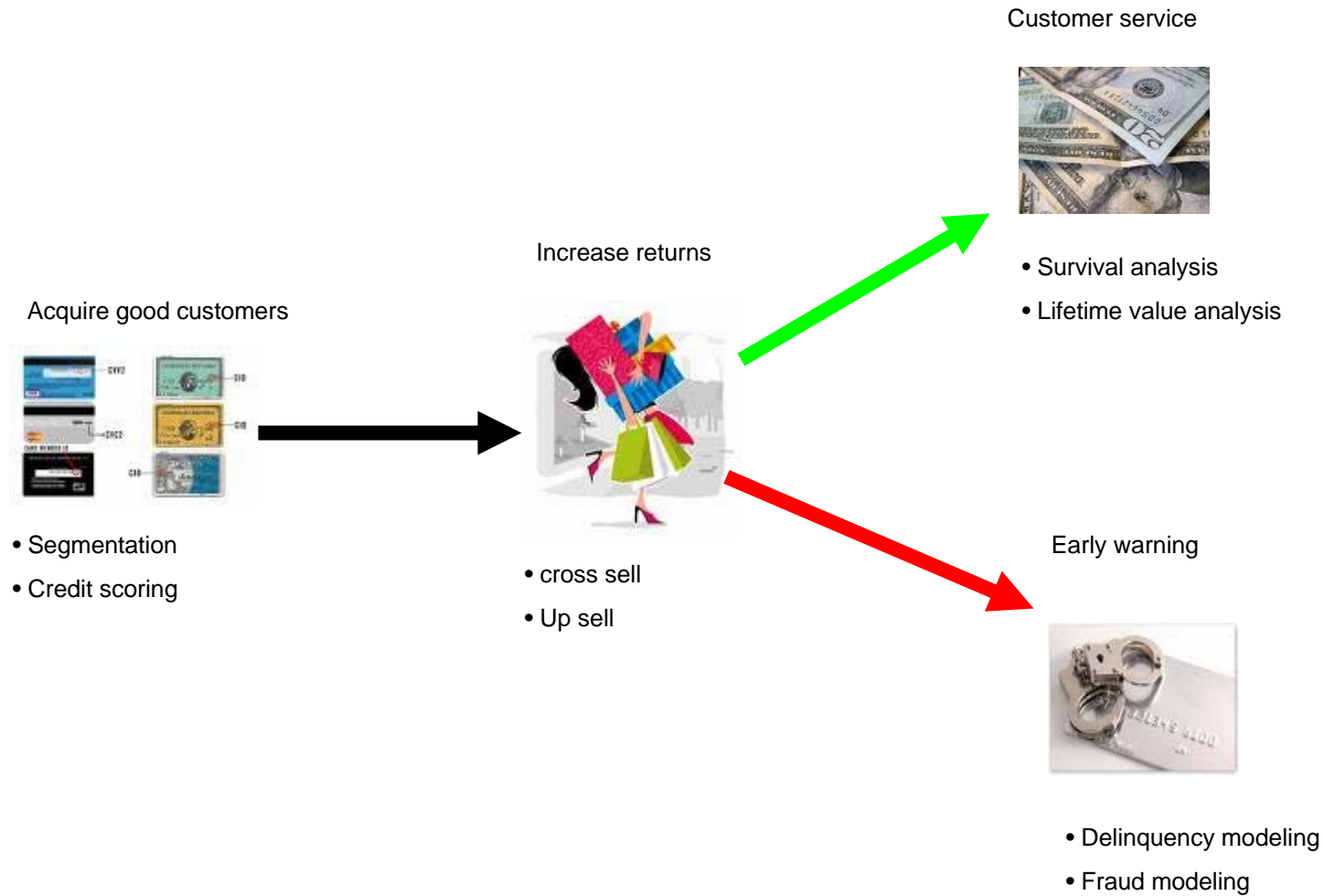
**HP, India**

# Background

- Credit risk management requires estimating future vulnerabilities & losses
- Criticality of predictive models in credit risk management
- Role of predictive modeling in asset groups having similar characteristics
- Statistical & data-driven approaches
- Sampling & algorithm selection issues



# Business Requirements for Predictive Modelling



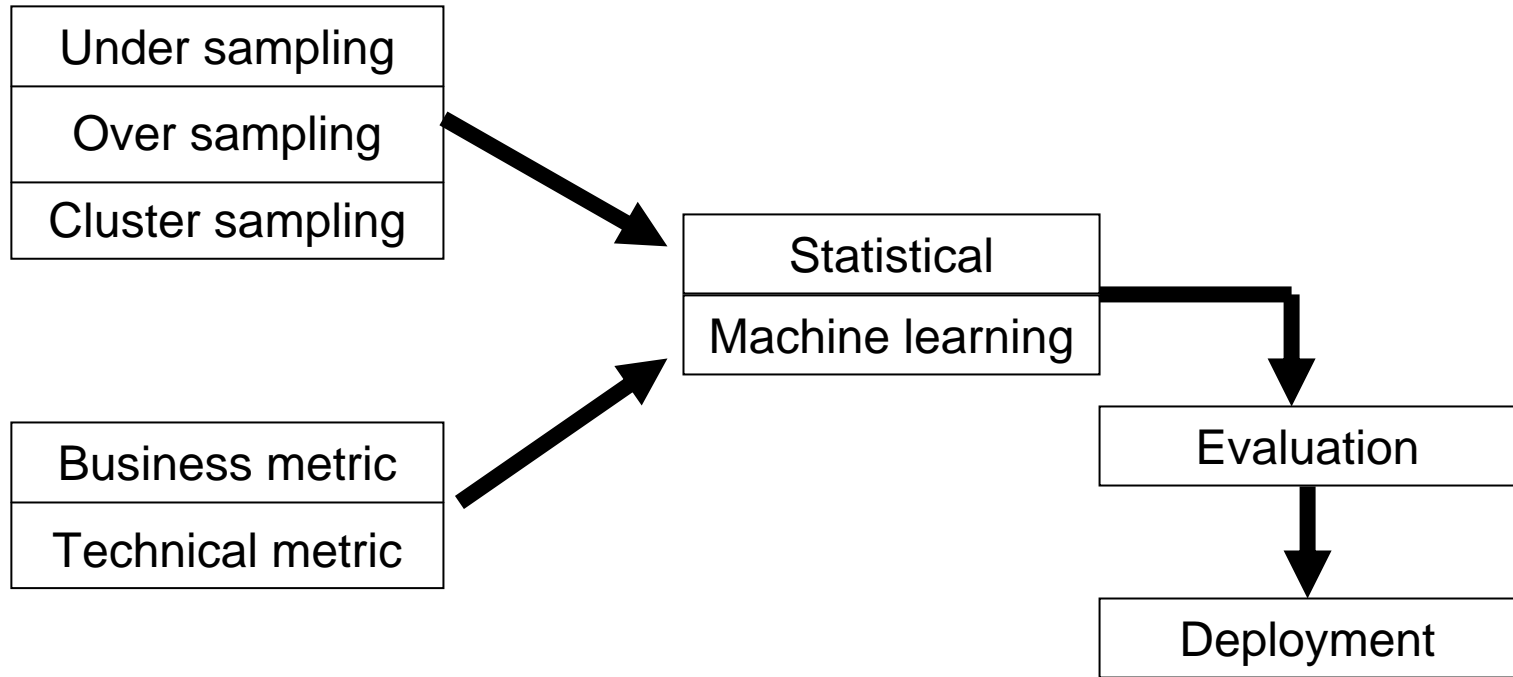


## Catch the Defaulter Early

- Before first 10 days from first use
- Before first 20 transactions
- Before 25% of credit limit utilization
- Limit the field monitoring to the likely default cases & save costs
- Minimize the probable loss

- Skewed/imbalanced dataset – straight forward modeling will result in poor performance
- Correlated features – results are too trivial for the domain experts
- Business requirement – Focus on defaulters prediction accuracy rather than on overall classification

# Default modeling – proposed solution



## **Under sampling**

Positive cases are retained and negative cases are randomly sampled to balance

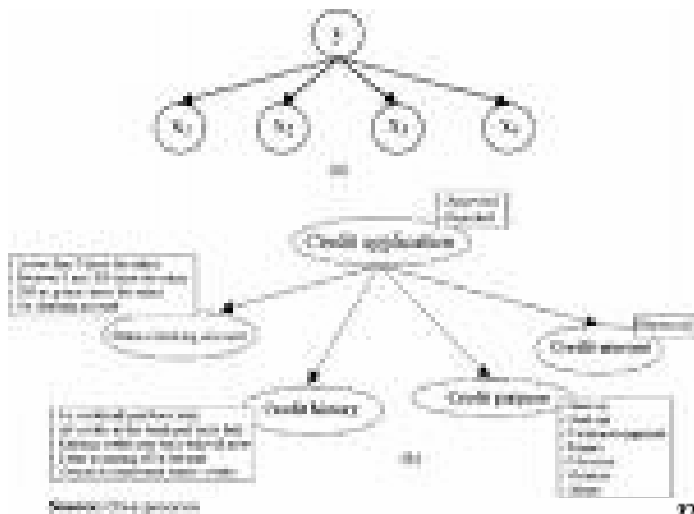
## **Over sampling**

Negative samples are over sampled to match and balance the positive case

## **Cluster sampling**

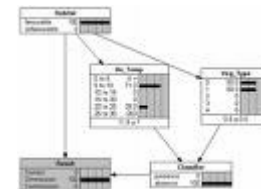
This is a modified under sampling. Positive cases are sampled with cluster prototypes to balance the negative cases.

## Naïve Bayes



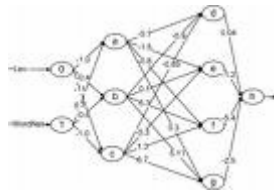
$$\text{classify}(f_1, \dots, f_n) = \text{argmax}_c P(C = c) \prod_{i=1}^n p(F_i = f_i | C = c)$$

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{parents}(X_i))$$



## Full Bayes

# Classifiers

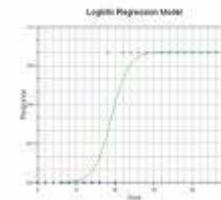


Neural nets (FBP)

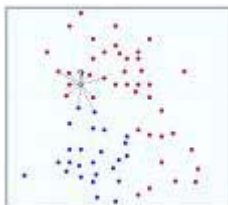
$$\hat{f}(x) = K \left( \sum_i w_i g_i(x) \right)$$

Logistic regression

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$



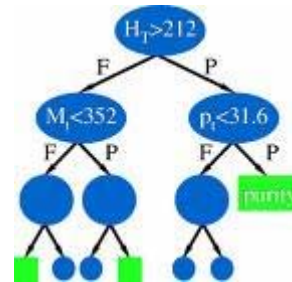
KNN



$$p(x|C_k) = \frac{K_k}{N_k V}$$

# Classifiers

## Decision trees



$$I_G(i) = 1 - \sum_{j=1}^m f(i, j)^2 = \sum_{j \neq k} f(i, j) f(i, k)$$

## Decision tables

NUM_TRAN_5	NUM_CASH_TRAN	CREDIT_LIMIT	REGION	DEFAULT
< 6	< 2000	5000	ANY	Yes (0.9)
< 2	< 100	10000	ANY	No (0.92)

## Decision rules

If NUM\_TRAN\_5 < 6 and  
NUM\_CASH\_TRAN < 2000  
Then Default (Prob. = 0.9)

## Data characteristics:

- Real world data set
- 120 fields, 20,000 records
- 10% default cases
- Pre processing
  - Cleaning
  - Imputing missing values
  - Sample preparation
- Validation
  - 10 fold cross validation

# Results

Table 1. Results for over-sampling the minority class for balancing

Sample plan →	Over sampling		
Algorithm ↓	CA	TP	TN
Simple Bayes	57.3	36.2	79.0
Complete Bayes	56.7	49.1	68.7
NBTree	57.0	43.4	70.5
Neural networks (FBP)	57.8	35.3	78.7
<b>Neural networks (RBN)</b>	59.2	<b>51.3</b>	61.2
Logistic regression	59.2	40.2	74.5
Decision trees	58.7	46.3	68.6
Decision rules	58.5	38.5	73.8
Decision tables	57.7	41.2	74.5
Support vector machines	56.9	38.1	68.5
Knn	56.5	39.2	67.0

# Results

Table 2. Results for under-sampling the majority class for balancing

Sample plan →	Under sampling		
Algorithm ↓	CA	TP	TN
Simple Bayes	60.5	37.5	81.5
Complete Bayes	59.2	50.1	68.2
NBTree	60.0	46.1	73.3
Neural networks (FBP)	59.8	36.2	81.9
<b>Neural networks (RBN)</b>	60.3	<b>55.4</b>	64.9
Logistic regression	61.1	44.2	75.1
Decision trees	60.2	48.7	71.2
Decision rules	59.0	40.5	77.9
Decision tables	60.8	43.4	77.4
Support vector machines	58.2	41.2	70.2
Knn	55.7	42.0	65.0

# Results

Table 3. Results for under-sampling the majority class for balancing with cluster prototypes

<b>Sample plan →</b>	<b>Over sampling</b>		
<b>Algorithm ↓</b>	CA	TP	TN
Simple Bayes	57.4	37.1	80.1
Complete Bayes	55.9	<b>48.2</b>	70.3
NBTree	57.6	43.5	71.7
Neural networks (FBP)	56.8	36.7	79.5
<b>Neural networks (RBN)</b>	60.3	<b>53.5</b>	60.5
Logistic regression	58.9	42.3	75.2
Decision trees	58.6	<b>47.6</b>	69.7
Decision rules	59.1	40.5	75.9
Decision tables	57.0	41.5	73.3
Support vector machines	57.2	39.7	69.4
Knn	57.1	40.6	62.6

### **General**

- The lift curve shows better prediction of default with model
- Both sampling and classifier seem to have effects on prediction performance

### **Sampling**

- Sample balancing has significant effect on performance improvement
- Balanced samples have higher accuracy
- Under sampling seems to be better than other methods
- Cluster sampling has not shown significant improvement despite the minimum information loss assumption

## **Classifier**

- Only few classifiers have performed well from the business perspective, even though classification accuracy of most of them are not significantly different
- RBF net with prior clustering and logistic classifier seems to be winner from the business perspective
- True negative prediction accuracy is better with simple classifiers like Bayes

- Defaulter modeling with business objectives are more acceptable by domain experts than classical prediction techniques
- Sampling in general and under sampling in particular will enhance the classification accuracy
- Classifier has to be selected based on the data profile and the business objective. Usual classification accuracy or lift can not be taken as performance indicators.
- It may be desirable to use more than one classifier to predict true positives and true negatives separately.

- Risk segmentation and outlier detection
- Performance improvement with multi classifier models
- Data profiling for performance and model selection

## References

Bharatheesh T.L, Iyengar S.S, “Predictive Data Mining for Delinquency Modeling”, . ESA/VLSI 2004: 99-105

Satchidananda S.S and Jay B.Simha, “Comparing the efficacy of the decision trees with logistic regression for credit risk analysis ”, SUGI Asia conference, Mumbai, India, 2006

Japkowicz, “The Class Imbalance Problem: Significance and Strategies,” in *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning*, (Las Vegas, Nevada), 2000. (18)

Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka: Practical machine learning tools and techniques with java implementations, *Proceedings of ICONIP/ANZIIS/ANNES'99 Int. Workshop: Emerging Knowledge Engineering and Connectionist-Based Info. Systems*, 1999