# 2019 Data Mining Cup Best Solution

Qihao Zhang, Yifan Zhu, Xingche Guo

Department of Statistics, Iowa State University

September 9, 2019

# Table of Contents

# Problem Introduction and Data Description

# Scenario

- **Data**: A self-checkouts data in retail collecting by handheld scanners
- **Domain Knowledge**: Approximate 5% discrepancy
- **Discrepancy**: Intentionally, or accidentally, or machine problem
- **Task**: Classify a half million scans in the test set as fraudulent or not fraudulent by building up a model on the training set
- **Evaluation**: Achieve the highest monetary profit on the test set

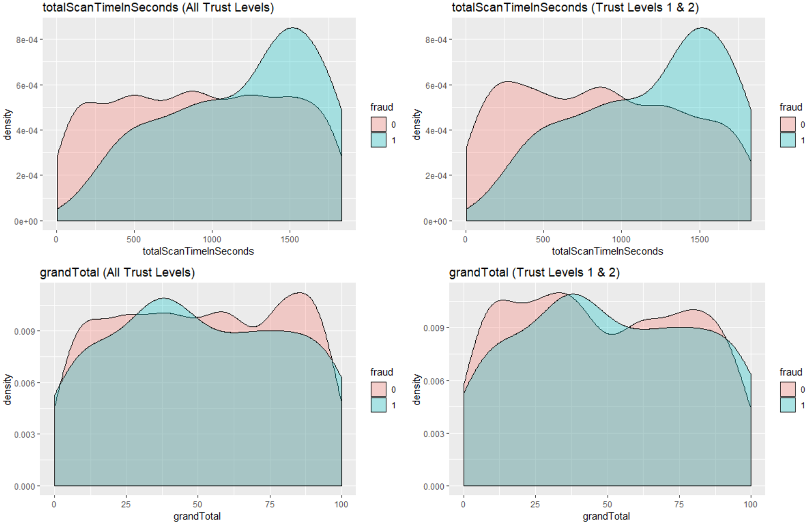
# Features

| | | |
|---|---|---|
| 1 | TrustLevel | How trustworthy |
| 2 | totalScanTimeInSeconds | How long for purchasing (Seconds) |
| 3 | grandTotal | How much spent ($) |
| 4* | lineItemVoids | number times of Void Scanning |
| 5* | scansWithoutRegistration | number times of Invalid Scanning |
| 6 | quantityModification | number times of error but legitimate scanning |
| 7** | scannedLineitemsPerSeconds | How fast of scanning (item/second) 10/2 |
| 8** | valuePerSecond | How fast of scanning ($/second) 3/2 |
| 9** | lineItemVoidPerPosition | Void Scanning/Legitimate Scanning 4/10 |
| 10* | itemTotal (New feature) | number times of Legitimate Scanning (total items) (Manually made $2\times7$) |
| 11 | fraud (Response) | fraud or not fraud (1 or 0) |

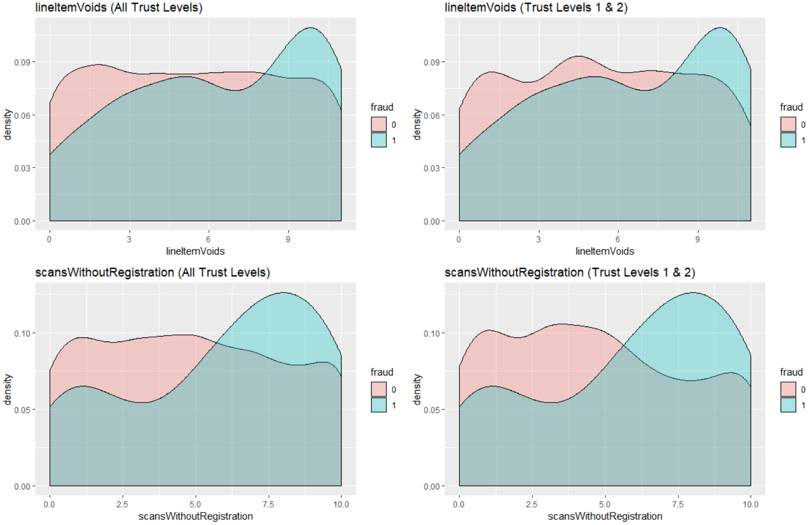
# Training Set

- Total 1879 scan observations and 9 original features
- 104 (5.5%) samples are frauds and 1775 (94.5%) samples are no frauds
- Denote fraud as 1 and no fraud as 0
- No fraud in the training set with TrustLevel 3, 4, 5, 6

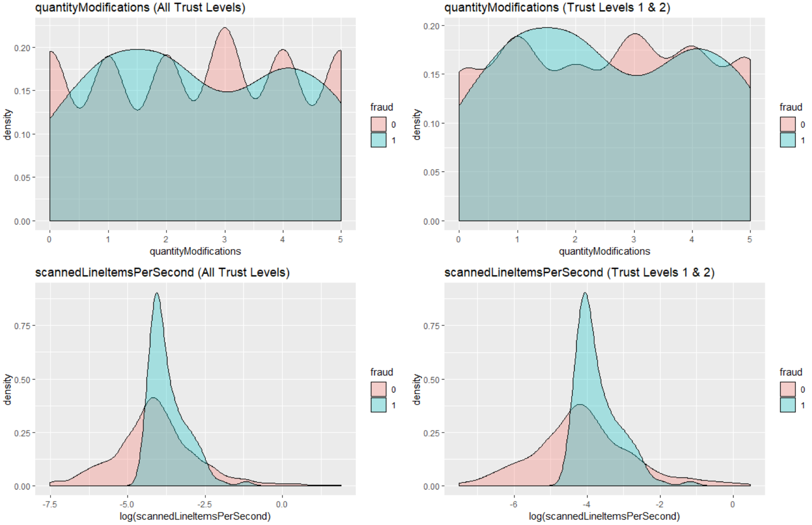
# Conditional Empirical Distribution $\hat{f}(x|y)$



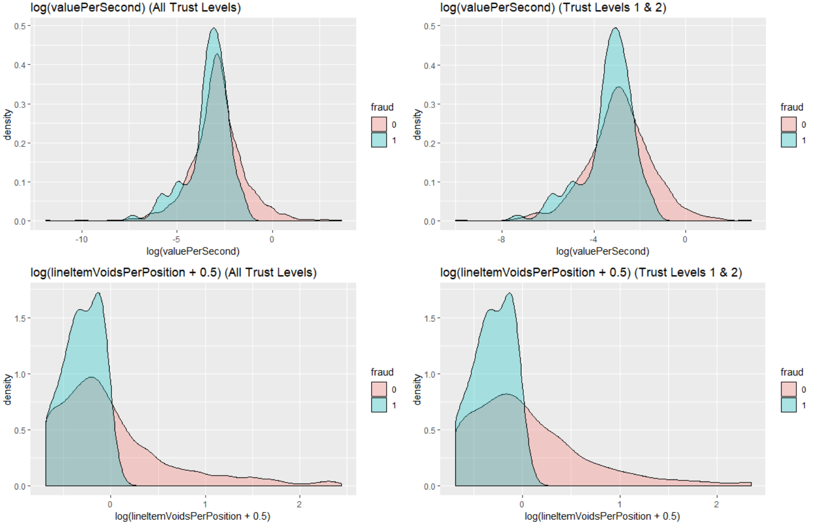
# Conditional Empirical Distribution $\hat{f}(x|y)$



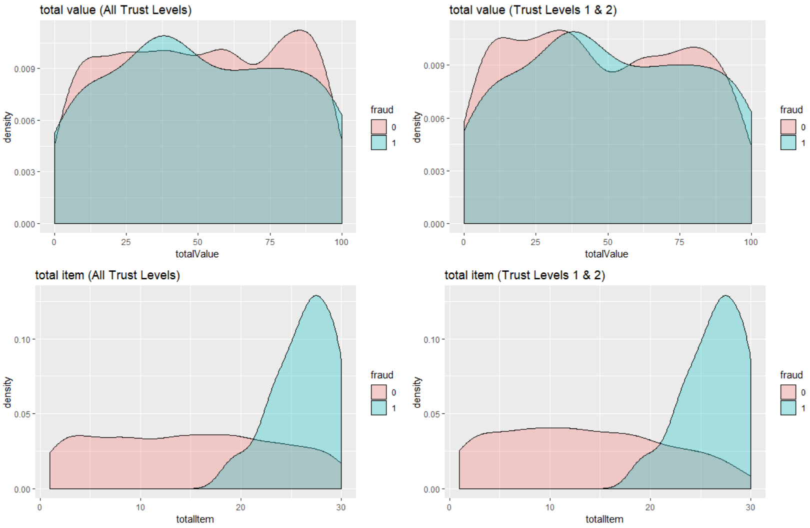
# Conditional Empirical Distribution $\hat{f}(x|y)$



# Conditional Empirical Distribution $\hat{f}(x|y)$



# Conditional Empirical Distribution $\hat{f}(x|y)$

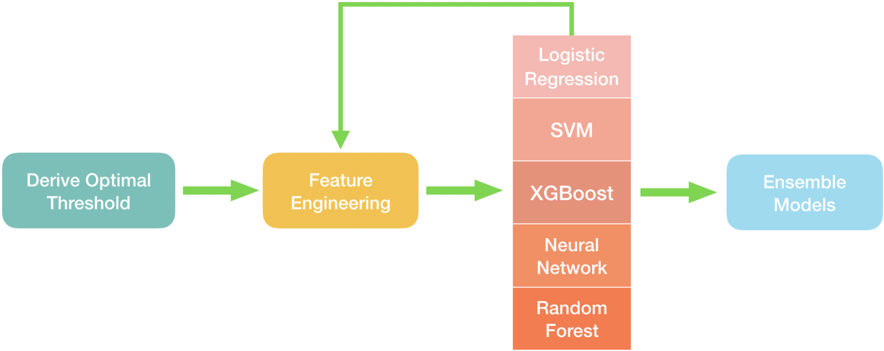


# Monetary Profit (Score function)

| Prediction | Actual value | | |
|---|---|---|---|
| | | 0 (no fraud) | 1 (fraud) |
| | 0 (no fraud) | 0 | -5 |
| | 1 (fraud) | -25 | 5 |

- The sum of the profit or score of all scans in the testing set is the monetary value of the submitted solution.
- We need to submit $0-1$ prediction instead of the predicted probability.
- Score Function:
  $S(y,\hat{y}) = 5 \times I(y=1,\hat{y}=1) - 5 \times I(y=1,\hat{y}=0) - 25 \times I(y=0,\hat{y}=1)$

# Modeling Procedure



# Evaluation Criteria and Cross-Validation Design

# Loss Function and the Optimal Threshold

- An unbalanced loss is given by the negative of the score function in this problem. Therefore we need an optimal decision rule for this loss.
- Let $y$ and $\hat{y}$ be the truth and prediction and be the vector of features.
- Denote $L(y, \hat{y}) = -S(y, \hat{y})$ as the loss function in this contest, we predict $\hat{y} = 1$ when:

$$\mathbb{E}[L(y, 1)|] < \mathbb{E}[L(y, 0)|],$$

By solving the inequality, we have:

$$p(1|) > \frac{L(0,1) - L(0,0)}{L(1,0) - L(0,0) - L(1,1) + L(0,1)} = \frac{5}{7}$$

Therefore, **fraud is detected when** $\hat{p}(1|) > 5/7$.

# The Oracle Bound

- With the optimal threshold, all the models hereby aim to find a good approximation of the conditional probability $\hat{p}(1|)$.

- For this training data set, we have 104 fraud 1 out of 1879 data points. Since we cannot classify every customer correctly as fraud or no fraud , then oracle upper bound of total score (when all are classified correctly) for this training data is

$$\sum_{i=1}^{n} S(y_i, \hat{y}_i) = 5 \times 104 = 520$$

# Model Performance Evaluation

- We use repeated cross-validation to evaluate the prediction ability of the models.
- For each repetition:
  1. Shuffle training data and split shuffled data set into k folds
  2. Fit the model using $k-1$ folds and predict on the remaining 1 fold according to the optimal threshold
  3. Calculate the score on each left-out fold
  4. Run thorough all $k$ left-out folds; add these scores and get a total score, i.e.

$$\sum_{i=1}^{n} S(y_i, \hat{y}_i^{CV})$$

- We repeat the cross validation 100 times with different partitions of the dataset and obtain 100 total scores. A good model should give a high total score and be robust to different partitions: the mean and variability of the 100 scores are used to evaluate a model

# Cross Validation Design

- Two ways to split the dataset:
  - (Random Split) randomly split the data into $k$ folds of about the same size
  - (Proportional Split) split the data into $k$ folds of about the same size such that the proportion of fraud in each fold is the same as the original data
- Number of folds used: $k = 5$ and $k = 10$
- The combination leads to 4 different designs of cross validation. Every model is evaluated under all 4 designs.

# "Best Linear" Model

For all subset of the original features + totalItem, based on our evaluation criteria, the best model we have now is the logistic regression model with 6 terms:

$$\textit{trustLevel} + \textit{totalItem} + \textit{lineItemVoids} + \textit{scansWithoutRegistration} + \textit{totalScanTimeInSeconds} + \textit{grandTotal}$$

The table below shows the negative of total score (we can call it total loss) $\sum_{i=1}^{n} L(y, \hat{y}_i^{CV}) = -\sum_{i=1}^{n} S(y_i, \hat{y}_i^{CV})$.

| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -313.05 | -309.75 | -304.65 | -306.5 |
| SD | 25.3 | 19.5 | 23.7 | 18.9 |
| Min | -365 | -320 | -345 | -365 |
| Q1 | -330 | -320 | -320 | -320 |
| Median | -310 | -310 | -310 | -310 |
| Q3 | -300 | -300 | -287.5 | -297.5 |
| Max | -235 | -240 | -240 | -260 |

# Feature Engineering and Machine Learning Model Selection

# Feature Engineering

To find good features $x$ that can separate 0 and 1

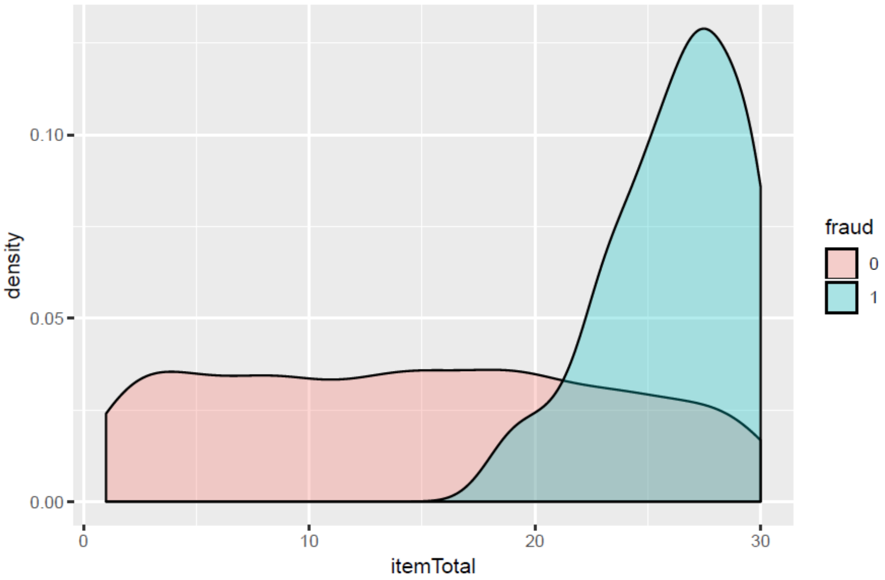
- Conditional distributions $f(x|y=0)$ and $f(x|y=1)$ have separable domains

- Conditional distributions $f(x|y=0)$ and $f(x|y=1)$ have different shapes

For example:

$$totalItem = scannedLineItemsPerSecond \times totalScanTimeInSeconds$$

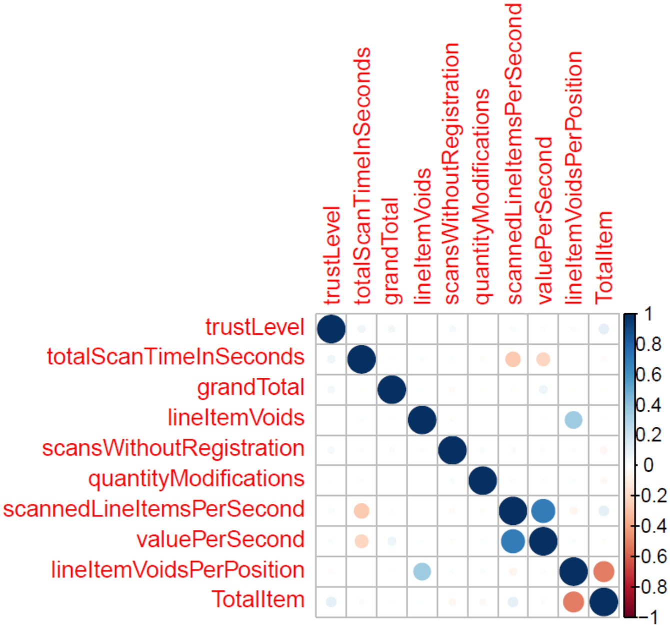
is an important feature made from the original features.


density
0.10
0.05
0.00
0
10
20
30
itemTotal
fraud
0
1

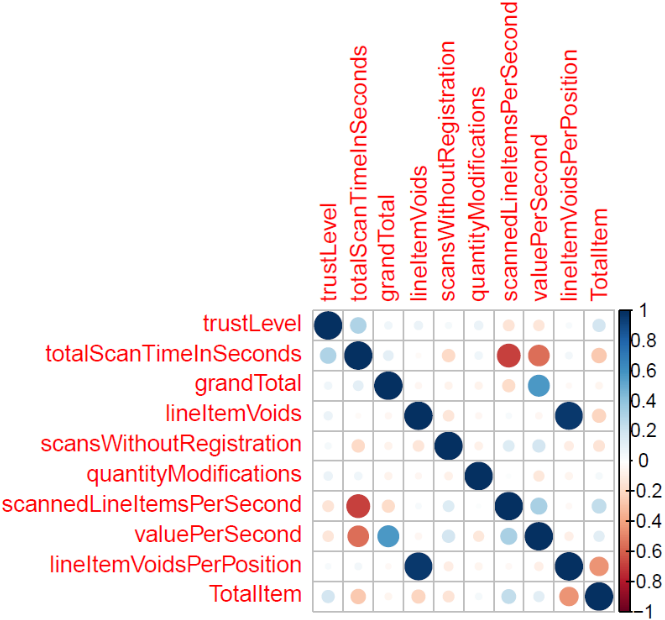
# Important Interaction Terms

- The correlation plot matrix is made separately for all data labeled "fraud" and "no fraud".
- Important interaction terms are added to the set of candidate features.

## no fraud:


trustLevel
totalScanTimeInSeconds
grandTotal
lineItemVoids
scansWithoutRegistration
quantityModifications
scannedLineItemsPerSecond
valuePerSecond
lineItemVoidsPerPosition
TotalItem
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

# fraud:


trustLevel
totalScanTimeInSeconds
grandTotal
lineItemVoids
scansWithoutRegistration
quantityModifications
scannedLineItemsPerSecond
valuePerSecond
lineItemVoidsPerPosition
TotalItem
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

# Model Selection

- Original features and potential good non-linear combination of the original features (e.g. totalItem) are added to the model
- Potential important interactions are added to the model
- Importance of features are obtained for each machine learning model, and unimportant features are removed
- All models are re-evaluated with only important features
- Logistic regression models turn to be the best using our evaluation criteria; other models tend to overfit

# The Final Method and Solutions

# Best Model So Far

Our best model so far is the logistic regression model with feature set:

$$trustLevel + totalItem + lineItemVoids + scansWithoutRegistration + totalScanTimeInSeconds + grandTotal + grandTotal \times valuePerSecond$$

| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -311.9 | -336.5 | -307.3 | -334.7 |
| SD | 47.9 | 27.5 | 41.7 | 28.6 |
| Min | -395 | -385 | -395 | -390 |
| Q1 | -347.5 | -360 | -335 | -357.5 |
| Median | -315 | -340 | -310 | -340 |
| Q3 | -285 | -325 | -275 | -315 |
| Max | -150 | -230 | -180 | -260 |

Compare to the best additive logistic model (with 6 additive terms), the best logistic interaction model:

- performs much better for 10-fold CV.
- has similarly mean scores and larger variance for 5-fold CV.

# Further Feature Engineering

Note that

$$grandTotal = totalScanTimeInSeconds \times valuePerSecond,$$

we can re-express the so far best model as:

$$trustLevel + totalItem + lineItemVoids + scansWithoutRegistration +$$
$$totalScanTimeInSeconds \times (1 + valuePerSecond + valuePerSecond^2)$$

Define $V := valuePerSecond$; $T := totalScanTimeInSeconds$; $f(V)$ to be some function of $V$. Then one possible underlying feature would be:

$$Interaction(T, V) = T \times f(V),$$

where in our best logistic model so far:

$$f_{(2)}(V) = 1 + V + V^2.$$

# Further Feature Engineering

$f_{(4)}(V) = 1 + V + V^4$.

| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -296.05 | -355.3 | -316.8 | -349.1 |
| SD | 242.5 | 34.5 | 71.9 | 50.93 |
| Min | -420 | -420 | -420 | -420 |
| Q1 | -350 | -377.5 | -350 | -370 |
| Median | -327.5 | -360 | -325 | -357.5 |
| Q3 | -292.5 | -335 | -295 | -335 |
| Max | 2040 | -215 | 110 | 65 |

Compare to $f_{(2)}(V)$, the logistic model with $f_{(4)}(V)$:

- performs even better for 10-fold CV.
- performs worse for 5-fold CV.

# Further Feature Engineering

$f_{(log)}(V) = 1 + log(V)$.

| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -328.9 | -337.5 | -326.6 | -339.2 |
| SD | 24.4 | 16.7 | 26.4 | 17.9 |
| Min | -375 | -365 | -375 | -385 |
| Q1 | -345 | -355 | -345 | -355 |
| Median | -330 | -332.5 | -330 | -340 |
| Q3 | -315 | -330 | -310 | -330 |
| Max | -230 | -295 | -240 | -285 |

Compare to $f_{(4)}(V)$, the logistic model with $f_{(log)}(V)$:

- performs much better for 5-fold CV (in terms of both mean and variance).

# Logistic Ensemble Models

## By Probability Mixture distribution

Define $\hat{\mathrm{P}}(y = 1|\boldsymbol{x}, \mathcal{F}_i)$ to be the fitted conditional probabilities by logistic regression using feature set $i$, denote $\hat{\omega}_i = \hat{\mathrm{P}}(\mathcal{F}_i)$, then:

$$\hat{\mathrm{P}}^{(en)}(y = 1|\boldsymbol{x}) = \sum_{i=1}^{d} \hat{\omega}_i \hat{\mathrm{P}}(y = 1|\boldsymbol{x}, \mathcal{F}_i), \quad \textit{subject to} \sum_{i=1}^{d} \hat{\omega}_i = 1.$$

- The ensemble model integrates the simple model (low fitting error, high model error) and complex model (high fitting error, low model error).
- Choose proper weights such that the ensemble model has smaller fitting error + model error.

# Logistic Ensemble Models

## By Probability Mixture distribution

Define:

$$baseLine = trustLevel + totalItem + lineItemVoids + scansWithoutRegistration,$$

A simple ensemble model with:
$baseLine + T$ and $baseLine + T \times (1 + V + V^2)$.

| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -352.8 | -354.85 | -346.2 | -358.3 |
| SD | 28.3 | 19.8 | 28.3 | 18.6 |
| Min | -420 | -400 | -410 | -400 |
| Q1 | -375 | -365 | -365 | -375 |
| Median | -355 | -355 | -345 | -360 |
| Q3 | -335 | -345 | -327.5 | -345 |
| Max | -280 | -290 | -285 | -305 |

The ensemble model performs much better than any single logistic models.

# Logistic Ensemble Models

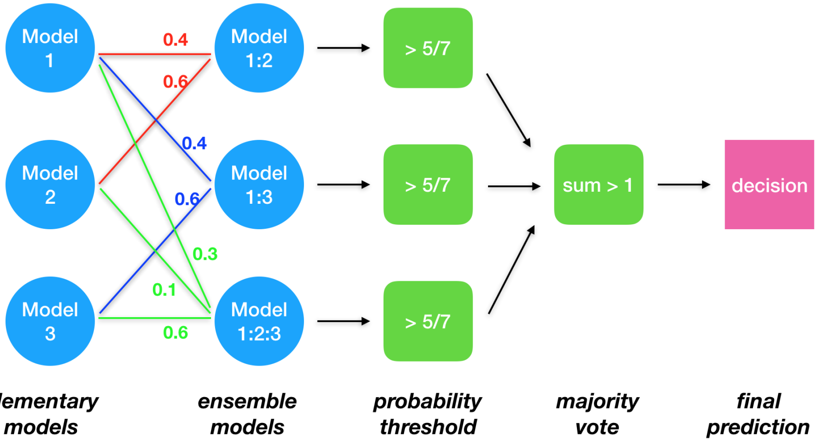
## By Probability Mixture distribution

Another ensemble model with:
$baseLine + T$ and $baseLine + T \times (1 + V + V^4)$.

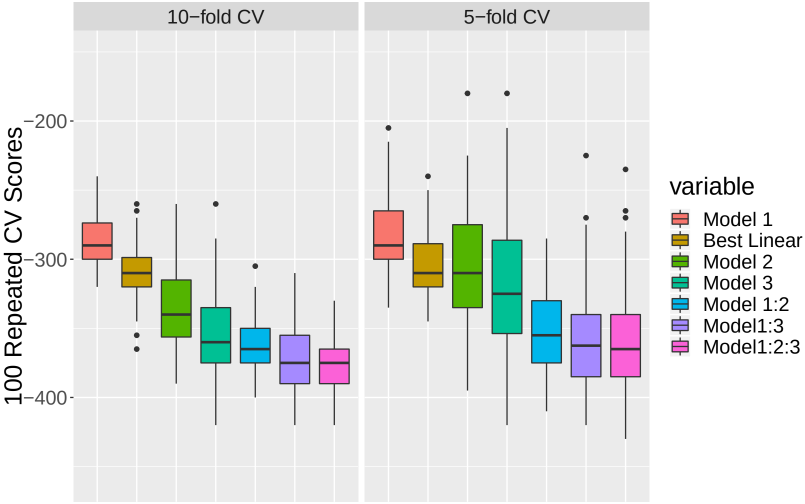
| | Random Split CV | | Proportional Split CV | |
|---|---|---|---|---|
| | 5 Fold | 10 Fold | 5 Fold | 10 Fold |
| Mean | -361.1 | -369.7 | -360.2 | -374.5 |
| SD | 34.5 | 27.1 | 31.2 | 20.9 |
| Min | -420 | -420 | -420 | -420 |
| Q1 | -390 | -390 | -385 | -390 |
| Median | -365 | -375 | -365 | -375 |
| Q3 | -345 | -352.2 | -340 | -360 |
| Max | -230 | -295 | -275 | -325 |

# Our Final Model (Logistic Ensemble Model)

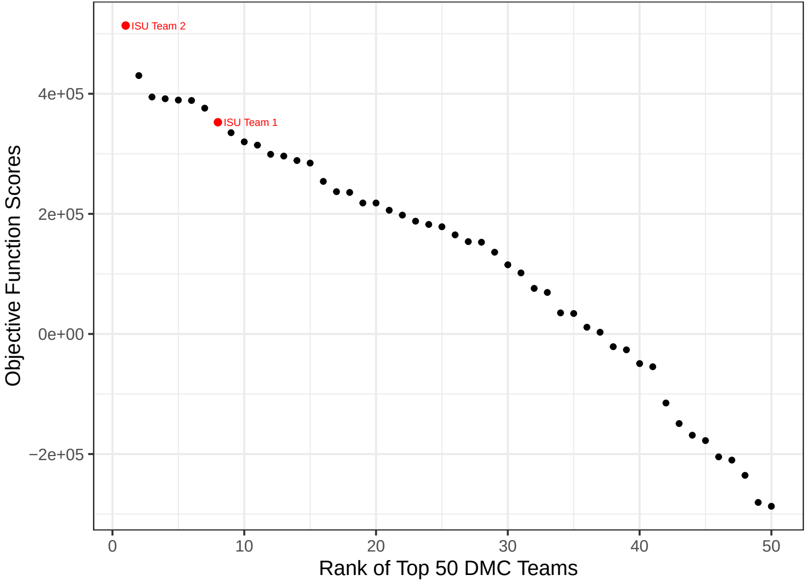
Feature Set 1: $baseLine + T$.
Feature Set 2: $baseLine + T \times (1 + V + V^2)$.
Feature Set 3: $baseLine + T \times (1 + V + V^{3.5})$.



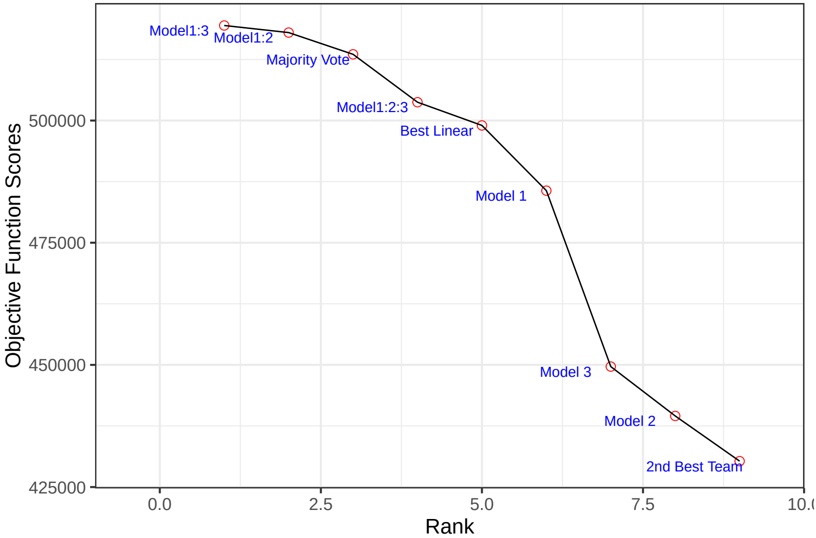
# Logistic Ensemble CV Results



# Our Final Solution vs Rest of Top 50 Teams



# All of Our Solutions vs Rest of 2nd Best Team



# Summary

# Key Ingredients Lead Us to Win

- Derive optimal threshold for unbalanced 2-class loss function.
- Use multiple evaluation criteria for model selection.
- Successful feature engineering.
- Use proper model ensemble method.
- Most importantly: we do not overfit the data.

# What We Learned from the Contest

- Simple models are more preferable for small datasets.
- Try to use a smaller fold Cross-Validation for a problem with small dataset.
- Fancy ML/DL methods do not guarantee you to win a data mining contest, spend more time on data.

# Advice for the Coming Year DMC

- Comprehend the task before going through the model details.
- Spend more time on data pre-study and feature engineering.
- Be organized (solve the problem step by step).
- Team work ( exchanging ideas and thoughts / time arrangement / managing file platform / responsible for teammates ) leads to win the contest.
- We believe you will have a good performance in the next year's contest as we did.

# Thank You!