Final Project: Methods for classification task with missing data

Hangyu Lin, Jichen Yang

Fudan University

18210980008@fudan.edu.cn
18210980017@fudan.edu.cn

June 21, 2019

Overview

Introduction

Problem

Evaluate the risk of car companies' dealers. The risk evaluation of the 515 dealers over 5 regions in Mar. 2019. The potential risk factors from two sources were provided

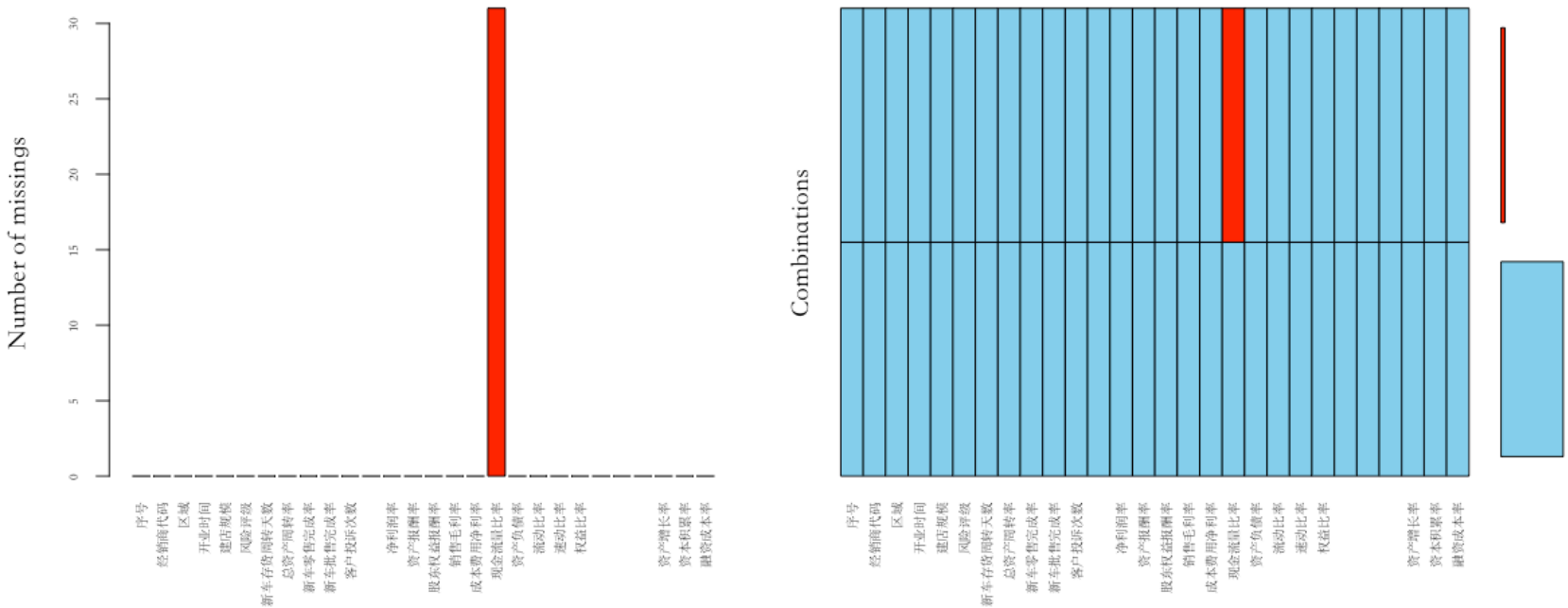
Inside accounting:

- 区域, 建店规模, 新车批售完成率, 客户投诉次数, 净利润率

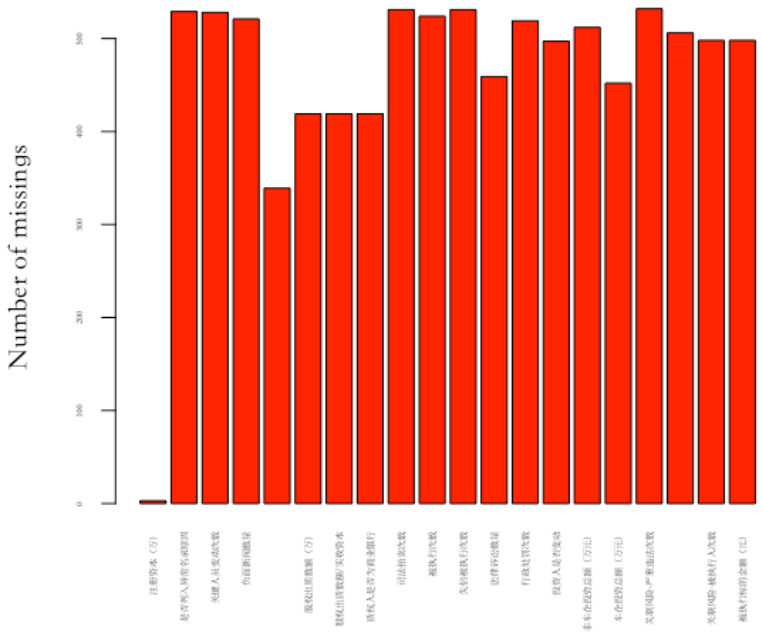
Outside commercial database:

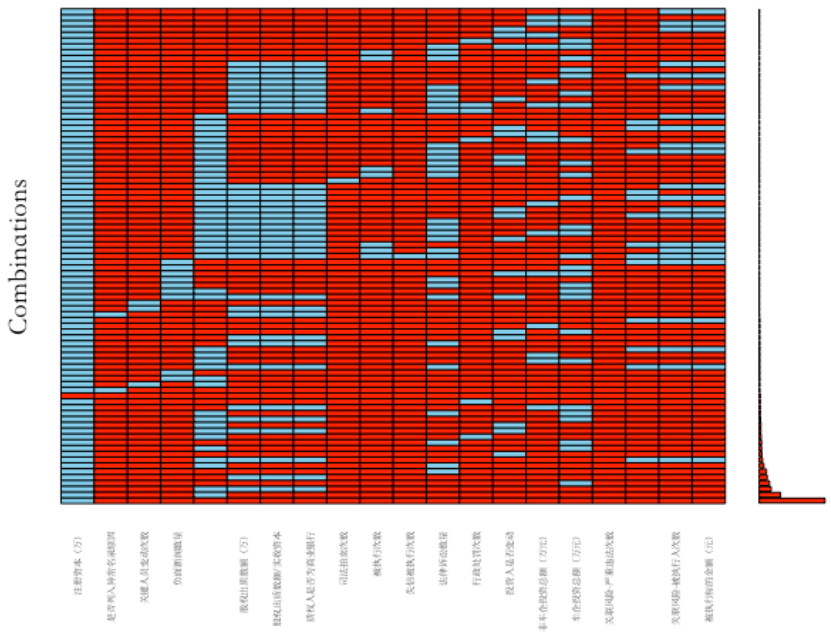
- 注册资本, 负面新闻数量, 投资人是否变动, 关联风险. 失信被执行人次数

Missing data: Inside accounting



Missing data: Outside commercial database





Methodology

Preprocessing: Missing Data

- Missing value represents 0, it's easy to complete.
  - 失信被执行次数, 法律诉讼数量, 行政处罚次数
- Low percentage of missing, we need some imputation method to complete it.
  - 车企投资总额, 现金流量比率
- The percentage of missing is large and the correlation with the predictor is very low, delete these variables.
  - 司法拍卖次数, 质权人是否为商业银行

We also delete the cases which do not have responses 风险评级.

Preprocessing: Specific Imputation Methods

- For continuous predictors, impute average value of the non-missing obervations.
- For categorical predictors, the imputed value is the category with the largest average proximity.

Preprocessing: Detect Outliers

The processing of outliers is divided into the following two steps.

1. The upper and lower bounds of some variables are given in the **range** table, and some observations exceed the upper and lower bounds. After filting, I find that some values above the upper and lower bounds are not outliers. So I adjusted the size of these values to make them equal to the upper and lower bounds. Then delete the remaining outliers.
2. Use statistical methods to test outliers in statistical sense.

Preprocessing: Detect Outliers

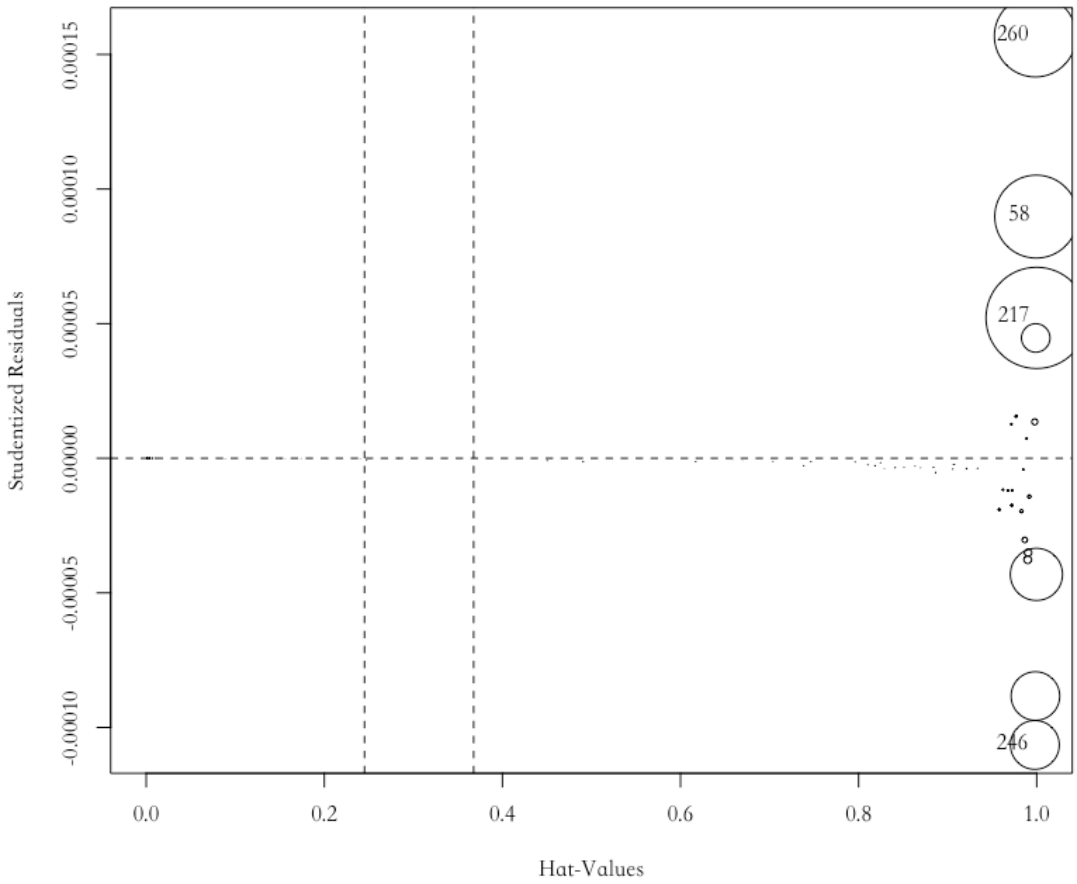
Leverage score measures how much each observation contributes to the model's prediction. The defination of leverage score as following:

$$h_{ii} = [\mathbf{H}]_{ii}$$

where $H = X(X^T X)^{-1} X^T$ is the projection matrix.
Cook's distance is a commonly used estimate of the influence of a data point when performing a generalized linear model. The formula for Cook's distance is:

$$D_i = \frac{\sum_{j=1}^{n} \left(\widehat{Y}_j - \widehat{Y}_{j(i)} \right)^2}{(p+1)\widehat{\sigma}^2}$$

Preprocessing: Detect Outliers



Logistic Regression: Select factors

Akaike information criterion (AIC)

$$AIC = 2k - ln(L) = 2k + nln(RSS/n) \tag{1}$$

The final formula is 风险评级 $\sim$ 资产增长率 $+$ 被执行人数.

Classification And Regression Tree (CART)

1. Start at the root node.
2. For each varible X_i, find the set S that minimizes the sum of the node impurities in the two child nodes and choose the split that gives the minimum overall X_i and S.
3. If a stopping criterion is reached, exit. Otherwise, repleat step 2 to each child node in turn.

Classification And Regression Tree (CART)

The key point in this kind of algorithms is the impurity function. In a node m, representing a region R_m with N_m observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{I}(y_i = k) \tag{2}$$

We will denote the class of a node as $k(m) = \arg max_k \hat{p}_{mk}$ There are several measures of node impurity include the following:

1. **Misclassification error**: $\frac{1}{N_m} \sum_{i \in R_m} \mathbb{I}(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$
2. **Gini index**: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
3. **Cross-entropy or deviance**: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$

Conditional Inference Tree(CTree)

1. Start at the root node.
2. For exisiting tree, test the global null hypothesis of independence between any of the m covariates and the response then selecting a covariate X_i with strongest association to response.
3. Choose the split that gives the minimum impurity overall X_i and S.
4. If a stopping criterion is reached, exit. Otherwise, repleat step 2 and 3.

Missing values and Surrogate splits

Unlike general deletion or imputation methods for solving missing data, CART and CTree can deal effectively with missing values through surrogate splits. Surrogate splits can be established by searching for a split leading to roughly the same division of the observations as the original split.

Boosting: Adaboost

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Boosting: Gradient Boost Decision Tree(GBDT)

$$F_0(x) = arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3}$$

$$F_m(x) = F_{m-1}(x) + arg\min_{h_m} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h_m(x_i)) \tag{4}$$

$$\tag{5}$$

Unfortunately, choosing the best function h at each step for an arbitrary loss function L is a computationally infeasible optimization problem in general. Therefore, we restrict our approach to a simplified version of the problem.

Boosting: GBDT and XgBoost

GBDT:

$$F_m(x) \approx F_{m-1}(x) + arg\min_{h_m} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i)) + g_i h_m(x_i) \tag{6}$$

$$\tag{7}$$

XgBoost:

$$F_m(x) \approx F_{m-1}(x) + arg\min_{h_m} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i)) + g_i h_m(x_i) + 1/2 h_i h_m(x_i) \tag{8}$$

$$\tag{9}$$

where $g_i = \partial_{F_{m-1}(x)} L(y_i, F_{m-1}(x_i)), g_i = \partial^2_{F_{m-1}(x)} L(y_i, F_{m-1}(x_i))$

Boosting: Gradient Boost Decision Tree(GBDT)

Input: training set $\{(x_i, y_i)\}_{i=1}^n$, a differentiable loss function $L(y, F(x))$, number of iterations M.

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

2. For m = 1 to M:
   1. Compute so–called *pseudo–residuals*:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

   2. Fit a base learner (e.g. tree) $h_m(x)$ to pseudo–residuals, i.e. train it using the training set $\{(x_i, r_{im})\}$
   3. Compute multiplier γ_m by solving the following one–dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i-1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

   4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

Experimental results and discussion

2-class classification: Logistic Regression

Confusion Matrix:

	Predicted True	Predicted False
Actual True	131	0
Actual False	3	8

Here we present the parameter of each variable and the stand error, we can use the stand error to construct the confidence interval. P-value is very small means that we can reject the null hypothesis that the parameter is 0.

	estimate	Std.	p-value
Intercept	-5.96546	1.30732	5.04e-06
资产增长率	10.97699	5.63691	0.051494
被执行人数	0.09537	0.02603	0.000249

3-class classification

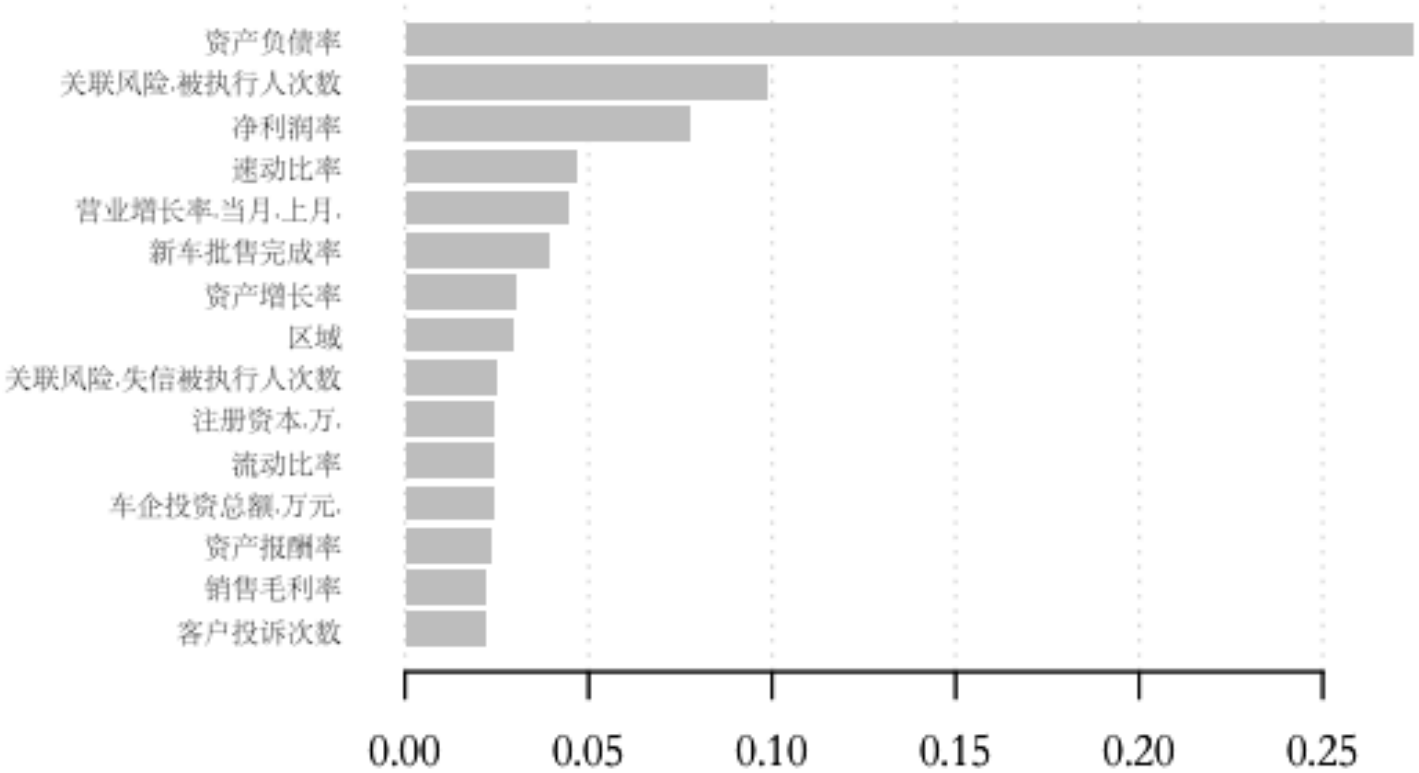
Logistic Regression:

	Predicted low	Predicted middle	Predicted high
Actual low	122	11	0
Actual middle	34	29	0
Actual high	2	4	2

XgBoost:

	Predicted low	Predicted middle	Predicted high
Actual low	120	13	0
Actual middle	19	44	0
Actual high	1	2	5

3-class classification: Select Factors



2-class classification: Tree-based Methods

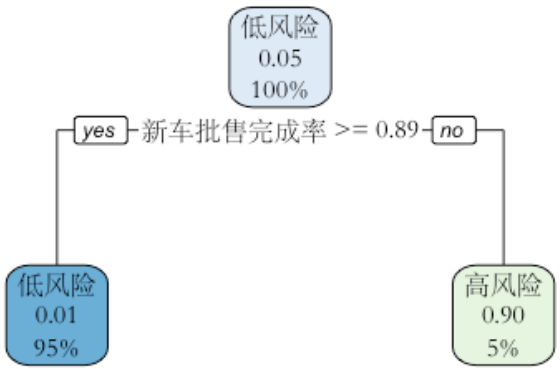
Method	CART		CTree	
	Low	High	Low	High
Actual Low	132/93%	4/3%	132/93%	4/3%
Actual High	0/0%	6/4%	0/0%	6/4%

Table: The confusion matrix for tree-based methods of 2-class classification with only inside accounting predictors. (left value means the number, right value is the percentage)

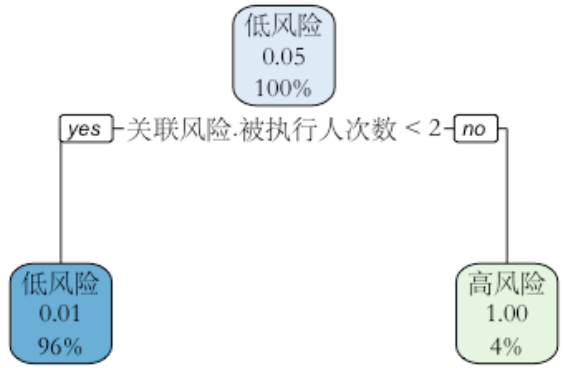
Method	CART		CTree	
	Low	High	Low	High
Actual Low	134/94%	1/1%	132/93%	0/0%
Actual High	0/0%	7/5%	1/1%	9/6%

Table: The confusion matrix for tree-based methods of 2-class classification with both inside and outside accounting predictors. (left value means the number, right value is the percentage)

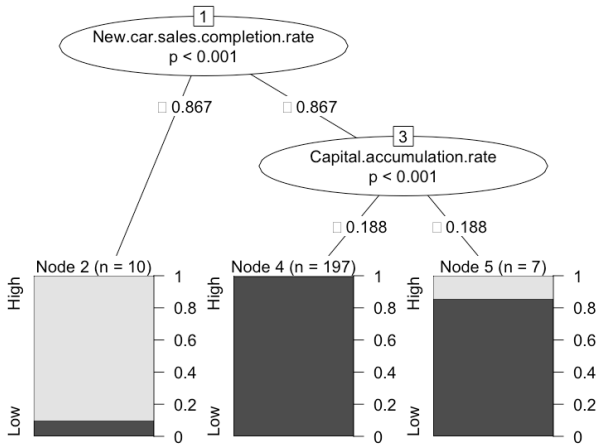
2-class classification: Tree-based Methods



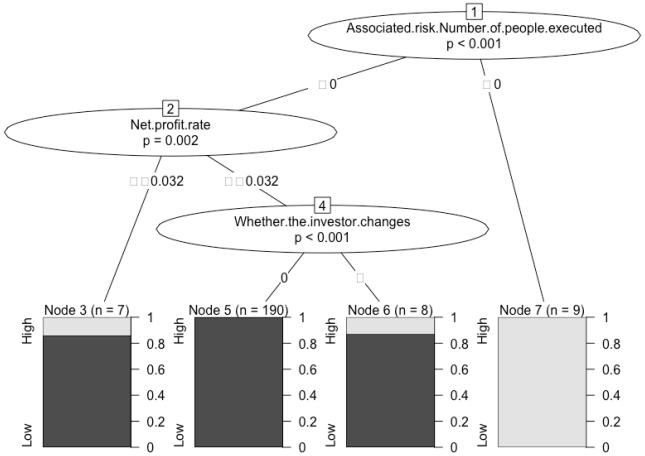
(a) CART with inside predictors on 2 class



(b) CART with inside and outside predictors on 2 class



(c) CTree with inside predictors



(d) CTree with inside and out-

3-class classification: Tree-based Methods

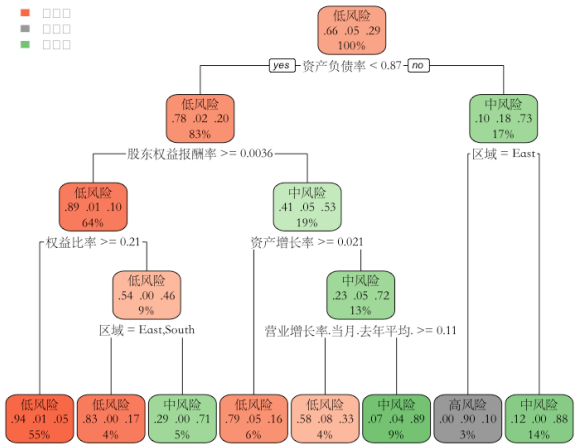
Method	CART			CTree		
	Low	Mid	High	Low	Mid	High
Actual Low	115/57%	23/11%	2/1%	105/52%	21/10%	1/0.5%
Actual Mid	19/10%	33/16%	3/1%	28/14%	32/16%	3/1%
Actual High	0/0%	2/1%	5/2%	1/0.5%	5/2%	6/3%

Table: The confusion matrix for tree-based methods of 3-class classification with only inside accounting predictors.

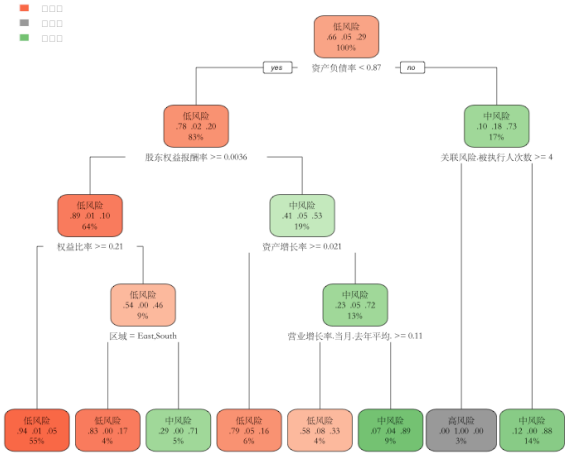
Method	CART			CTree		
	Low	Mid	High	Low	Mid	High
Actual Low	120/59%	20/10%	1/0.5%	99/50%	6/3%	0/0%
Actual Mid	19/10%	35/17%	5/3%	40/20%	48/24%	4/2%
Actual High	0/0%	0/0%	2/1%	0/0.5%	1/0.5%	4/2%

Table: The confusion matrix for tree-based methods of 3-class classification with both inside and outside accounting predictors.

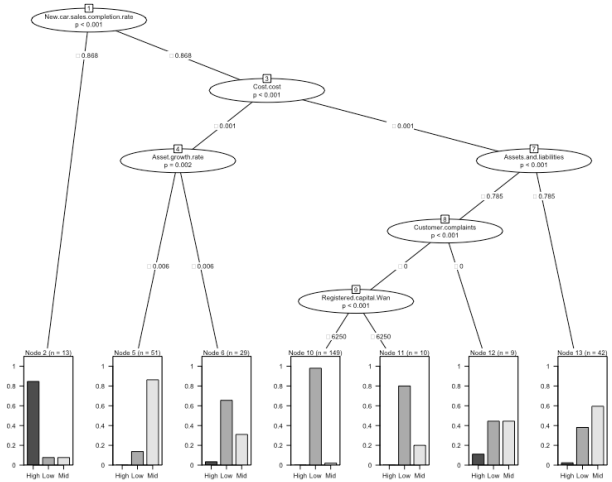
3-class classification: Tree-based Methods

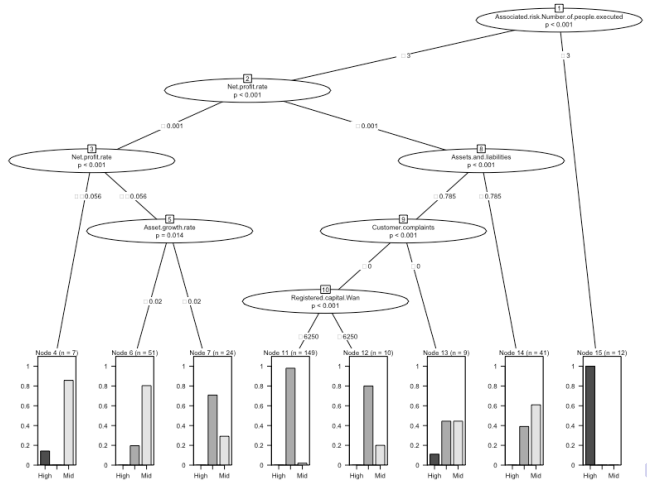


(a) CART with inside predictors on 3 class



(b) CART with inside and outside predictors on 3 class





References

References

Breiman L. (2017)

Classification and regression trees[M]

Routledge

Hothorn T, Hornik K, Zeileis A.

Unbiased recursive partitioning: A conditional inference framework[J]

Journal of Computational and Graphical statistics, 2006, 15(3): 651-674.

Breiman Chen T, Guestrin C.

Xgboost: A scalable tree boosting system[C]

Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.

Thanks