Contents

1 DESCRIPTION

1.1 Title

1.2 Field of The Invention

This invention involves in the field of cash loan system and determines whether a borrower is qualified for the loan.

1.3 Background

”Cash loan” means small cash loan business, which is a kind of consumer loan business for applicants. It has convenient and flexible loan and repayment process, as well as real-time approval and fast-acquisition. Platform participants of cash loans can be divided into private departments, banking departments, listing departments, state-owned departments and venture capital departments.

1.3.1 The Advantage of Cash Loan

The biggest advantage of cash loan is that the verification process is more lenient than the bank and that the target population is wider. The users of cash loans are generally low-income or unstable young groups, often dismissed by banks when they encounter emergencies. Cash loans do not require mortgage and can often help borrowers get through short-term difficulties. In addition, the approval speed for accessing the smart automatic approval system is relatively fast and brings a lot convenience. Borrowers can use their own credit as a guarantee, and there’s no need to hypothecate something. Cash loans can be borrowed repeatedly within the credit line, and can be borrowed at any time. And the credit rank of users with good credit will continue to increase.

1.3.2 The Disadvantage of Cash Loan

From a lender’s perspective view, it can provide exorbitant profit, but also there are several disadvantage. First of all, the way cash loan makes profit leads to high risk. Unlike credit card companies charge sellers to make profit, cash loan lenders gain most profit from those who cannot pay their debt in a short-time and keep paying high interest. At the meantime, there is not enough constraint to make sure cash loan borrowers pay their debt. Usually they use their personal ID and information as the mortgage instead of real valuable goods. In some situations, borrowers are either too poor to pay the debt or indifferent about their reputation anymore.

2 SUMMARY

To determine whether an individual is qualified for the loan, we implement the following steps to solve the problem. Firstly, we preprocess the data by normalizing and dealing with the missing data. The process of normalization makes data in the same order of magnitude so that features will have the same extend of impact on the results and avoid data of large orders of magnitude having too much impact on the outcome, thus making the outcome inaccurate. As for the missing data, we use the mean value to replace the data. Secondly, we label the specific column '3d' as the attribute of classification. Thirdly, we use the features and attribute of the train data to establish the decision tree classification. Then, we construct the random forest classification to make the classification better. Lastly, we optimize the main parameters to match the model and make the outcome more accurate.

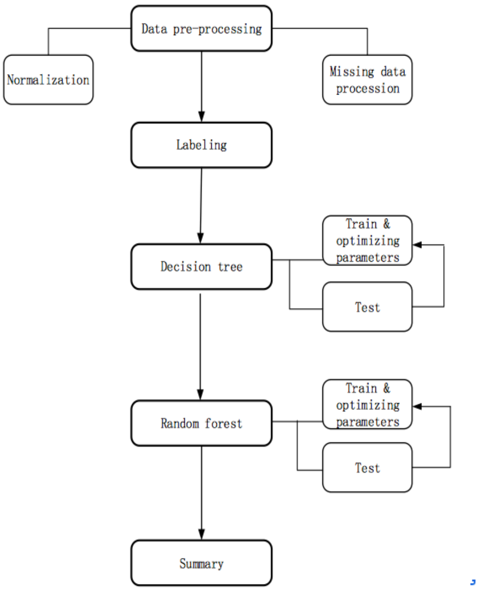


Figure 1: Flow Chart

3 DESCRIPTION OF DRAWING

4 DESCRIPTION OF PREFERRED EMBODIMENT

4.1 The Algorithm of Cash Loan Model

- **Decision tree**

  Decision tree is a data mining tool which builds tree-like classification or regression models to handle categorical or numerical data. It starts with a root node and breaks down into two or more branches, which are called decision nodes. Each decision node has its own branches and the tree keeps growing until it reaches a classification or decision. The final nodes are called leaf nodes. Thanks to this tree form structure, the results of a decision tree can be easily understood.

  A decision tree is built top-down from the root node which contains all the decisions, to leaf nodes which contain instances with similar values (also known as homogenous). The homogeneity of a sample is calculated using entropy, which is defined as:

  $$H(T) = -\sum_{i=1}^{J} p_i log_2 p_i$$

  where J is the number of classes and p_i is the percentage of each class. Entropy of zero means the sample is completely homogeneous and entropy of one implies the sample is equally divided. When building a decision tree, the entropy of the parent node is first calculated. Then the dataset is split on the different attributes. The new entropy of the child node is calculated as the weighted sum of entropy for each branch:

  $$H(T|a) = \sum_a p(a) \sum_{i=1}^{J} -Pr(i|a) log_2 Pr(i|a)$$

  To determine which attribute to be used to split the dataset, new entropy after splitting is subtracted from the entropy before the split, the result is called information gain:

  $$IG(T, a) = H(T) - H(T|a)$$

The attribute that has the largest information gain is used as the decision node to divide the dataset by its branches. This process is repeated until the branch has zero entropy, and that branch would be a leaf node.

- **Random Forest**

  Random forest is a relatively new machine learning model. The classic machine learning model is a neural network with a history of more than half a century. Neural network predictions are accurate, but it is computationally intensive. In the 1980s, Bre iman et al. invented the classification tree algorithm.

  It uses the bootstrap resampling technique to randomly extract k samples from the original training sample set N to generate a new training sample set, and then generate a new training sample set according to the self-service sample set. The k classification trees constitute a random forest, and the classification result of the new data is determined by the score formed by voting of the classification tree. Its essence is an improvement of the decision tree algorithm, which combines multiple decision trees.

  Each classification tree in the random forest is a binary tree, and its generation follows the principle of top-down recursive splitting, that is, the training set is divided from the root node in turn; in the binary tree, the root node contains all training data, according to the principle of minimum purity of node. The root node is split into left and right nodes, which respectively contains a subset of the training data, and the nodes continue to split according to the same rule until the branch stop rule is satisfied and the growth stops. If the classification data on node n is all from the same category, then the node's purity $I(n) = 0$.

4.2 Procedure

- **Step1: Data Acquisition**

- **Step2: Date Preprocessing**

  A data set is usually in rows and columns. Each row, in particular, can be interpreted as a single client that the cash loaning enterprise has. The columns represents different features of each client. Among all the columns there is a 3d column that indicates the true label for each client.

The first step of data preprocessing is dealing with missing data. Every set of data might contain missing data. Since they do not carry any helpful information themselves, missing datas would disturb further classification unless we dispose them appropriately. When the missing datas are scarce, we may manually delete the “bad” rows and classify the rest. However, this scenario rarely happens in reality when missing datas are usually intense or hard to be recognized in a big data set. Removing rows with missing datas arbitrarily would eventually results in huge missing information. Hence a method that not only identifies missing values but could somehow fix them is more preferable. Still, to what extend the missing datas are to be fixed depends on the values that already exist. We define a data preprocessing function which calculates the mean values of the columns. Each missing data would be replaced by the corresponding mean value. Since mean values are calculated with all existing data in a certain feature column, they are pretty representative for all data and could fill missing values convincingly.

Next comes data normalization. After we generate a relatively comprehensive data set with our mean value function, one would notice that the data varies significantly among different feature columns. Consequently, classification work is going to be painstaking once the range of values gets too large. The logic follows that the importance of each feature is not determined by the size of values it contains. For example, the feature “age” might only vary between 20 and 80, whereas feature “credit” might have values that are few times of those under the feature “age”. However, we know that the feature “credit” is not necessarily that much more important than “age”. Indeed, different features are not always comparable to each other. What we are really concerned with is that given a particular feature (i.e. credit), which person performs better than the others relatively. In order to eliminate potential misconstrue by our model, we acquire data normalization through rescaling the values under our feature columns into a range of 0 to 1. The idea is to transform datas into ratios that measure the relative differences of each individual. In the mean time, normalization dramatically narrows calculation and thus boost data processing with only a few lines of codes.

- **Step3: Training and optimization**

- Decision tree

Now that we have preprocessed our data, decision tree model is ready for the training. The 3d column which demonstrates true labels will be compared with predicted labels by our model. Therefore, the data will first be divided into variable train_y that contains 3d column and train_x that contains the remainder. We feed train_x to the model that will later provide us with corresponding labels of prediction.

- Random forest

We have already predicted the labels using our decision tree model but the outcome isn't accurate enough so we establish the random forest classification to solve this problem. Variable train_y contains the label '3d' and train_x contains the remainder. Then, repeat the step in part 4.2 and compare the true labels in column '3d' with the labels we predicted using this classification.

The main parameters of this classification have been listed below:

Parameter Description

Parameter	Descriptions
n_estimators	the number of decision tree
bootstrap	whether the sample set is sampled with put back to build the tree.
oob_score	whether to use out-of-pocket samples to evaluate the quality of the model.
max_depth	maximum depth of decision tree
criterion	the partition criteria of the node.

- Optimization

Actually the classification has already fitted our data set much more. However, we can still do some optimization by changing the main parameters of random forest classification, including n_estimators, that is, how many decision trees we establish in the random forest and max depth of each decision tree.

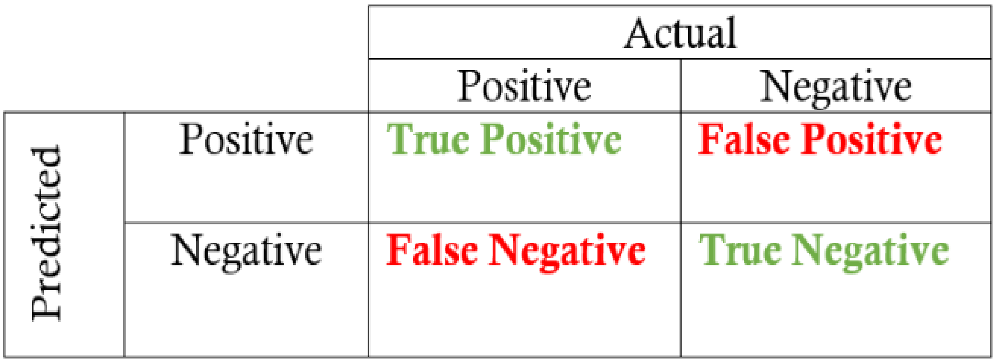
In the process of optimization, we first consider n_estimators in our model. According to some relevant materials, we set the range of n_estimators as 100 to 200. The output of our code indicates that when n_estimators is 150, the indexes we utilize to measure the classification attain the best condition.

Moving on, we add another parameter, max depth, to the code, while the range is set as 6 to 10. With two loop in our code, we can acquire the best result considering the two parameters. The final output is presented in the following picture.

- **Step4: Testing**

  In the process of testing, we utilize the preprocessed testing data to examine several the performance of our classifications. The principles of testing decision tree classification and random forest classification are quite similar. Therefore we only list the procedure of testing decision tree classification and that of testing random forest classification is in the similar fashion.

  We have labeled the features and attributes of the test data in the preprocessing stage. The features are regarded as test_x and will be applied to our decision tree classification and random forest classification. The model generates a vector called y_pre which indicates the result of our prediction. By comparing test_y with y_pre, we can measure the behavior of the model. The measurement standards we utilize include ROC_AUC, $f1$ score, precision and recall, while ROC_AUC is a major standard that we need to consider. ROC is the abbreviation for receiver operating characteristic curve, while AUC represents area under the ROC curve. The ROC curve is plotted as true positive rate versus false positive rate, the biggest values of whom are both 1. It behaves as a convex curve above the diagonal of the coordinates. The more AUC is close to 1, the more accurate our model is. The idea of other standards is explained as follows.



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 2: Confusion Matrix

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Figure 3: $f1$ Score

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives} \qquad precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Figure 4: Recall and Precision

With the testing, it is obvious that the behavior of random forest classification is much more better than decision tree classification.

figure of result

So we choose random forest theory to establish our model.

By optimization, the classification become increasingly fitter for our data.

figure of result