COMPARING DECISION TREE AND RANDOM FOREST DATA MINING METHODS, THE IMPACT ON THE PREDICTION RESULTS OF BANK CUSTOMER CREDIT CLASSIFICATION

E.I. Gubin, PhD, Associate Professor
Yubo Jin, student gr. 8PM0I
Tomsk Polytechnic University
E-mail: yuybo2@tpu.ru

Introduction

Data mining is the process of extracting information and knowledge that people do not know in advance but that has potential usefulness from a large amount of incomplete, noisy, fuzzy, and random actual data. Generally speaking, the results of data mining do not require completely accurate knowledge, but rather a general trend. As far as specific applications are concerned, data mining is the process of using various analytical tools to discover relationships between models and data in massive data sets, and these models and relationships can be used to make predictions. One of the primary risks faced by banks is credit risk, of which loan risk is the main element. This paper aims to screen credit customers according to the personal information on the finance dataset, finding the right person (the good customer).

Research methods

Using classification techniques in data mining, customers can be divided into different categories. The target variable of the customer's personal data we obtained is "GB", which means 1 is good, 0 is bad. It can be seen that this is a classification problem. We use decision trees and random forests in our classification technology to build credit risk models. The steps we need to do: 1) Data reading and preprocessing. 2) Model building. 3) Parameter optimization. 4) Model prediction and feature importance. 5) Visualize decision trees and random forests. 6) Compare the classification accuracy of decision trees and random forests. By comparing the two data mining classification methods, we analyze the reasons for the differences and select the most suitable and accurate model.

Results

First, use Python to read the raw data in CSV format, convert it into dataframe format, and view the dataset description. After that, clean the data, fill in the missing value with the mean of the same category, delete the noise data with excessive error, check that there are no duplicate values, and perform digitalization of the data in string format.

Here we select the prepared data sheet, which contains the credit information of 3000 people. We define the rest of the data, except "GB" as x (feature) and the categorical variable "GB" as y (target). Usually, 75% of the data set is used as the training set and the remaining 25% as the test set. Next, use the sklearn libraries to build decision tree and random forest models, respectively, and use the grid search method for parameter optimization.

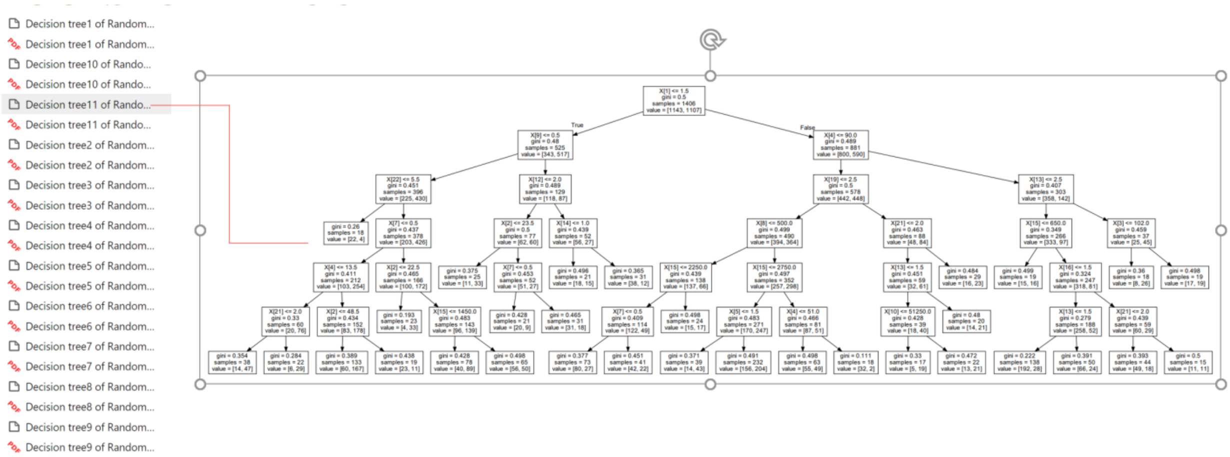
In Table 1, showing the importance of different features, the larger the value, or the closer it is to 1, the more important it is. On the contrary, the closer it is to 0, the less important it is. Here we will arrange them in reverse order.

Table 1. The importance of features

Feature	Importance
Age	0.227958
TMJOB1 (Time at Job)	0.101802
INCOME	0.077459
cards_ (Credit Cards int format)	0.072145
PERS_H (Time at Job)	0.069979
CASH (Requested cash)	0.065912
EC_CARD (EC_card holders)	0.049334
INC (Salary)	0.044683
prof_ (Profession)	0.040487
NMBLOAN (Num Mybank Loans)	0.039799

Samples: Total number of samples, Values: bad values/ good values.

We need to know from decision trees when to stop: One way of doing this is to set a minimum number of training inputs to be used on each leaf. Another way is to set the maximum depth of your model. Maximum depth refers to the length of the longest path from a root to a leaf. Here we choose one of the decision trees in the random forest for display.



Decision tree1 of Random...
Decision tree1 of Random...
Decision tree10 of Rando...
Decision tree10 of Rando...
Decision tree11 of Rando...
Decision tree11 of Rando...
Decision tree2 of Random...
Decision tree2 of Random...
Decision tree3 of Random...
Decision tree3 of Random...
Decision tree4 of Random...
Decision tree4 of Random...
Decision tree5 of Random...
Decision tree5 of Random...
Decision tree6 of Random...
Decision tree6 of Random...
Decision tree7 of Random...
Decision tree7 of Random...
Decision tree8 of Random...
Decision tree8 of Random...
Decision tree9 of Random...
Decision tree9 of Random...
X[1] <= 1.5 gini = 0.5 samples = 1406 value = [1143, 1107]
True
False
X[9] <= 0.5 gini = 0.48 samples = 525 value = [343, 517]
X[4] <= 90.0 gini = 0.489 samples = 881 value = [800, 590]
X[22] <= 5.5 gini = 0.451 samples = 396 value = [225, 430]
X[12] <= 2.0 gini = 0.489 samples = 129 value = [118, 87]
X[19] <= 2.5 gini = 0.5 samples = 578 value = [442, 448]
X[13] <= 2.5 gini = 0.407 samples = 303 value = [358, 142]
gini = 0.26 samples = 18 value = [22, 4]
X[7] <= 0.5 gini = 0.437 samples = 378 value = [203, 426]
X[2] <= 23.5 gini = 0.5 samples = 77 value = [62, 60]
X[14] <= 1.0 gini = 0.439 samples = 52 value = [56, 27]
X[8] <= 500.0 gini = 0.499 samples = 490 value = [394, 364]
X[21] <= 2.0 gini = 0.463 samples = 88 value = [48, 84]
X[15] <= 650.0 gini = 0.349 samples = 266 value = [333, 97]
X[3] <= 102.0 gini = 0.459 samples = 37 value = [25, 45]
X[4] <= 13.5 gini = 0.411 samples = 212 value = [103, 254]
X[2] <= 22.5 gini = 0.465 samples = 166 value = [100, 172]
gini = 0.375 samples = 25 value = [11, 33]
X[7] <= 0.5 gini = 0.453 samples = 52 value = [51, 27]
gini = 0.496 samples = 21 value = [18, 15]
gini = 0.365 samples = 31 value = [38, 12]
X[15] <= 2250.0 gini = 0.439 samples = 138 value = [137, 66]
X[15] <= 2750.0 gini = 0.497 samples = 352 value = [257, 298]
X[13] <= 1.5 gini = 0.451 samples = 59 value = [32, 61]
gini = 0.484 samples = 29 value = [16, 23]
gini = 0.499 samples = 19 value = [15, 16]
X[16] <= 1.5 gini = 0.324 samples = 247 value = [318, 81]
gini = 0.36 samples = 18 value = [8, 26]
gini = 0.498 samples = 19 value = [17, 19]
X[21] <= 2.0 gini = 0.33 samples = 60 value = [20, 76]
X[2] <= 48.5 gini = 0.434 samples = 152 value = [83, 178]
gini = 0.193 samples = 23 value = [4, 33]
X[15] <= 1450.0 gini = 0.483 samples = 143 value = [96, 139]
gini = 0.428 samples = 21 value = [20, 9]
gini = 0.465 samples = 31 value = [31, 18]
X[7] <= 0.5 gini = 0.409 samples = 114 value = [122, 49]
gini = 0.498 samples = 24 value = [15, 17]
X[5] <= 1.5 gini = 0.483 samples = 271 value = [170, 247]
X[4] <= 51.0 gini = 0.466 samples = 81 value = [87, 51]
X[10] <= 51250.0 gini = 0.428 samples = 39 value = [18, 40]
gini = 0.48 samples = 20 value = [14, 21]
X[13] <= 1.5 gini = 0.279 samples = 188 value = [258, 52]
X[21] <= 2.0 gini = 0.439 samples = 59 value = [60, 29]
gini = 0.354 samples = 38 value = [14, 47]
gini = 0.284 samples = 22 value = [6, 29]
gini = 0.389 samples = 133 value = [60, 167]
gini = 0.438 samples = 19 value = [23, 11]
gini = 0.428 samples = 78 value = [40, 89]
gini = 0.498 samples = 65 value = [56, 50]
gini = 0.377 samples = 73 value = [80, 27]
gini = 0.451 samples = 41 value = [42, 22]
gini = 0.371 samples = 39 value = [14, 43]
gini = 0.491 samples = 232 value = [156, 204]
gini = 0.498 samples = 63 value = [55, 49]
gini = 0.111 samples = 18 value = [32, 2]
gini = 0.33 samples = 17 value = [5, 19]
gini = 0.472 samples = 22 value = [13, 21]
gini = 0.222 samples = 138 value = [192, 28]
gini = 0.391 samples = 50 value = [66, 24]
gini = 0.393 samples = 44 value = [49, 18]
gini = 0.5 samples = 15 value = [11, 11]

Fig. 1. Visualize one of the decision trees in a random forest

Compare the performance metrics scores of the two models in Table 2:

Table 2. Get classification performance metrics

Performance metrics	Decision tree Score	Random forests Score
Accuracy	0.57333	0.67066
Precision	0.63788	0.68238
Recall	0.54653	0.69796
f1-score	0.58868	0.69008

It can be seen that the classification of random forests is better than that of decision trees because random forest is based on the combination of multiple weak classifiers of decision trees into a strong classifier.

Conclusion

A single decision tree can achieve complete accuracy or a root mean square error on the training set, which is impossible for random forests, but the performance of random forests on the test set is significantly better than that of decision trees. Avert overfitting through random forests and achieve better generalization performance. The accuracy of random forests is 67.066%, significantly better than the accuracy of decision trees, at 57.333%. So, we finally chose the random forest model.

References

1. Onesmus Mbaabu. (09.04.2020) Introduction to Random Forest in Machine Learning [Electronic resource]. – URL: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning
2. The application of big data in the financial industry [Electronic resource]. – URL: http://c.biancheng.net/view/3736.html
3. Prashant Gupta. (18.05.2017) Decision Trees in Machine Learning [Electronic resource]. – URL: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
4. Huang Shan, Gubin E. Data cleaning for data analysis. Томск, 2018г. - С. 387-389.