CREDIT CARD DEFAULT PREDICTION

Team Members:

1. Chandan Kumar Raxit

2. Deepak Kumar Jena

Abstract

Aiming at the problem that the credit card default data of a financial institution is unbalanced, which leads to unsatisfactory prediction results, this paper proposes a prediction model based on *k*-means SMOTE and BP neural network. In this model, *k*-means SMOTE algorithm is used to change the data distribution, and then the importance of data features is calculated by using random forest, and then it is substituted into the initial weights of BP neural network for prediction. The model effectively solves the problem of sample data imbalance. At the same time, this paper constructs five common machine learning models, KNN, logistics, SVM, random forest, and tree, and compares the classification performance of these six prediction models. The experimental results show that the proposed algorithm can greatly improve the prediction performance of the model, making its AUC value from 0.765 to 0.929. Moreover, when the importance of features is taken as the initial weight of BP neural network, the accuracy of model prediction is also slightly improved. In addition, compared with the other five prediction models, the comprehensive prediction effect of BP neural network is better.

1. Problem Statement

Can we reliably predict who has is likely to default? If so, the bank may be able to prevent the loss by providing the customer with alternative options (such as forbearance or debt consolidation, etc.). We will use various machine learning classification techniques to perform my analysis.

2. Data

Source, classic dataset from UC Irvine's machine learning repository: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Target: Did the customer default? (Yes=1/Positive, No=0/Negative)

Features:

1. Credit Limit: Amount of the given credit (in dollars): it includes both the individual consumer credit and his/her family (supplementary) credit

2. Sex (1=male; 2=female)

3. Education (1=graduate school; 2=university; 3=high school; 4=other)

4. Marital Status (1=married; 2=single; 3=others)

5. Age (years)

6. History of past payment: The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above

7. Amount of bill statement (dollars) for past 6 months

8. Amount of previous payment for the past 6 months

3. Introduction

We can see that the problem of category imbalance is mainly solved from the following two perspectives: the first perspective is to balance the data by changing the number of samples. This method can also be divided into three aspects. On the one hand, it is to improve the oversampling method. On the other hand, it is based on the principle of under sampling to change the data distribution. On the third hand, it is the method of combining oversampling and under sampling. The second perspective is to improve the classifier algorithm to improve the prediction performance of the model and at the same time use relevant evaluation indicators to evaluate the prediction results. Under normal circumstances, since under sampling will lose information, oversampling is the most widely used technique, and smote is the more common method. However, we have found that most scholars cannot reduce the imbalance between and within the sample categories at the same time when using the improved version of the smooth method, and the applicability of the improved version of the classifier is also limited. Therefore, this paper proposes an improved version of the smooth algorithm with better applicability, which combines the *k*-means algorithm. This method clusters all samples using the *k*-means unsupervised learning algorithm, finds clusters with more samples in the minority class, and then uses the smote method that synthesizes new samples in the cluster to change the data distribution. It can not only reduce the imbalance between the categories but also reduce the imbalance within the categories. At the same time, it combines the BP neural network method to predict the credit card default situation to help the bank to identify credit card risks effectively.

4. Basic Theory

• PCA

The main idea of the principal component analysis (PCA) method is to transform the *n*-dimensional feature variable through the coordinate axis and the origin to form a new *m*-dimensional feature (usually, *m* is less than *n*) [11]. This *m*-dimensional feature is also called principal component. Its essence is to replace a series of related sample features with newly generated comprehensive features that are irrelevant to each other. When analyzing the data, you can set the cumulative variance ratio determination factor in advance.

• Feature Importance Calculation of Random Forest

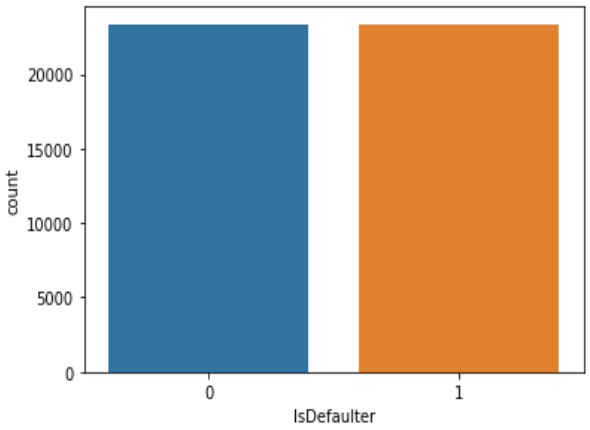
Random forest is a relatively basic machine learning algorithm, which is widely used in predictive analysis [12], data labeling [13], tag ranking [14], feature importance calculation [15], and other fields. The principle of the algorithm is as follows: using bootstrap method to randomly construct *n* decision trees, each decision tree is split and pruned and finally combined to form a random forest. In this paper, random forest is used to calculate feature importance, which is used as the initial weight of BP neural network.

• BP Neural Network

The prediction model used in this paper is the BP neural network algorithm, which is a feed-forward neural network for error backward update. It is often used for bank risk analysis [16], geological disaster monitoring [17], image and handwritten digit recognition [18, 19], and other fields. BP neural network consists of three parts: input layer, middle layer, and output layer. In the model, data samples enter the input layer through a weighted combination of different weights, then pass through the middle layer, and finally get the result from the output layer. Different weights and activation functions make the output of the model very different.

5. K-means SMOTE Algorithm

We know that smote is a method for synthesizing new samples and solving data imbalance proposed by Chawla et al. [20] and is widely used in various fields. Smote is an improved method of random oversampling technology. It is not a simple random sampling, repeating the original sample, but a new artificial sample generated by a formula. But the smote algorithm will also increase the imbalance between the positive and negative classes of the sample to a certain extent. Therefore, according to the problem of imbalance of credit card sample categories, this paper uses an improved smote algorithm called *k*-means SMOTE algorithm.

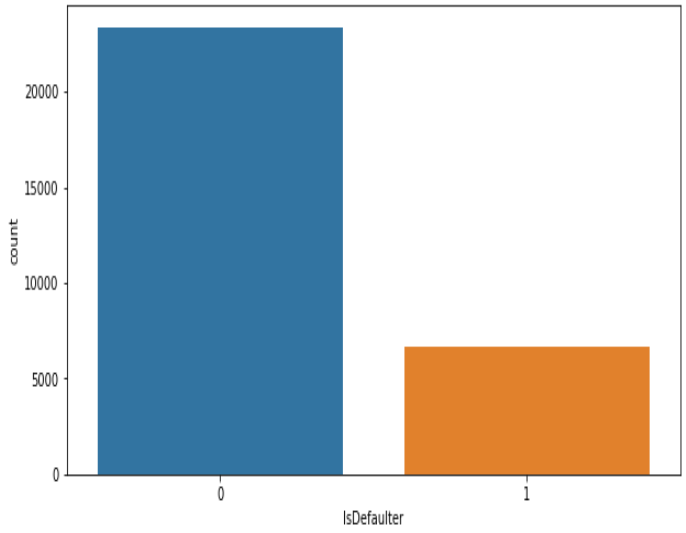


This algorithm can reduce the imbalance between categories on the one hand and reduce the imbalance within categories on the other hand. In this experiment, we first cluster all samples (30,000), then use *k*-means method to filter clusters with more minority categories, select clusters with more minority categories after filtering, and finally perform smote oversampling in the filtered clusters.

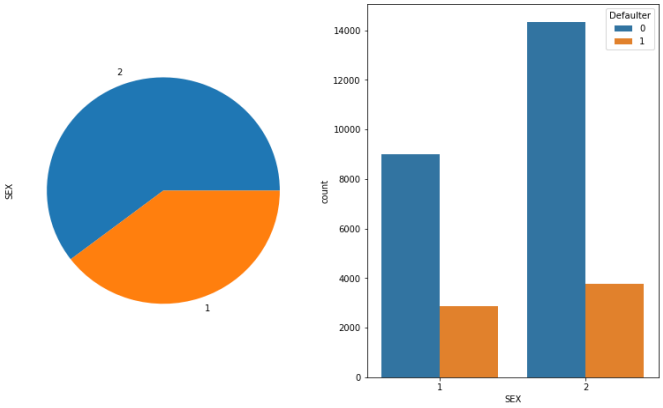
6. Experimental Data and Preliminary Analysis

- **Preliminary Analysis of Data**

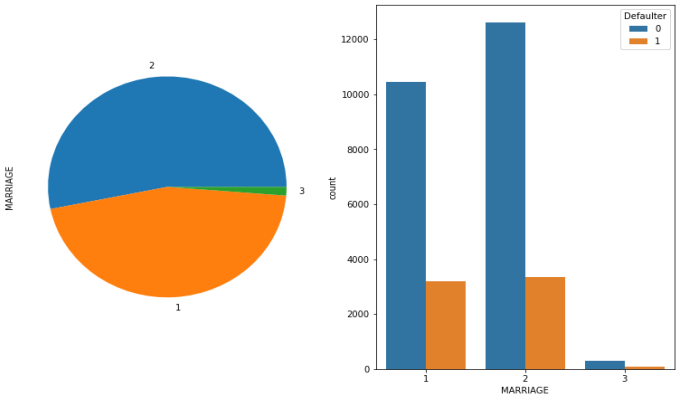
Distribution of target classes is highly imbalanced, non-defaults far outnumber defaults. This is common in these datasets since most people pay credit cards on time (assuming there isn't an economic crisis).



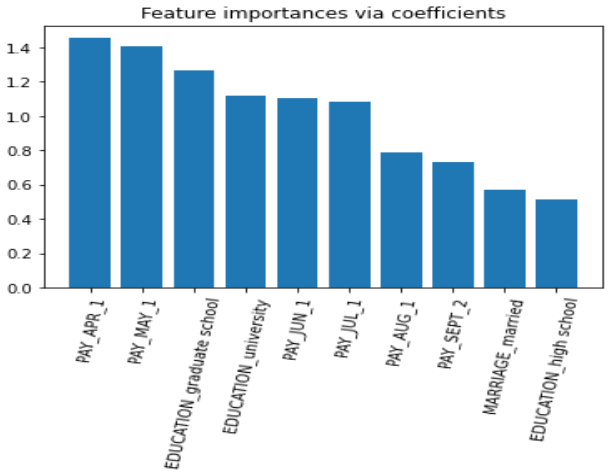
Credit Limit by Sex. The data is evenly distributed amongst males and females.



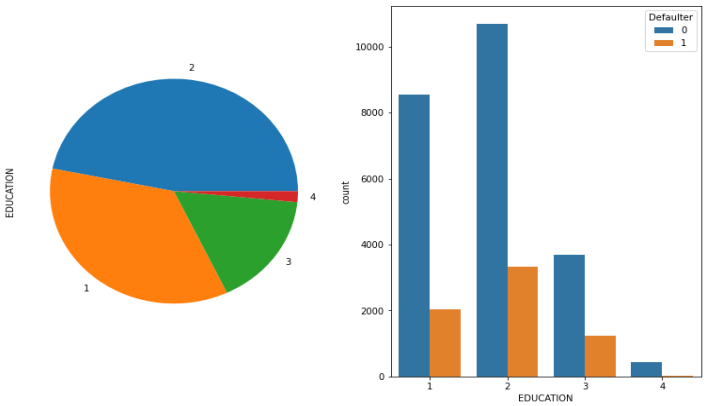
Marriage, age, and sex. The dataset mostly contains couples in their mid-30s to mid-40s and single people in their mid-20s to early-30s



7. Model Prediction and Comparative Analysis.

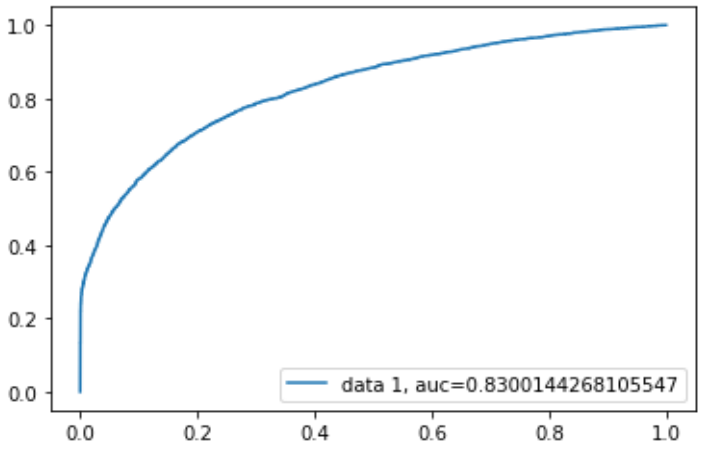


Credit Limit by Education. The data is evenly distributed amongst males and females.



- **Model Evaluation Method**

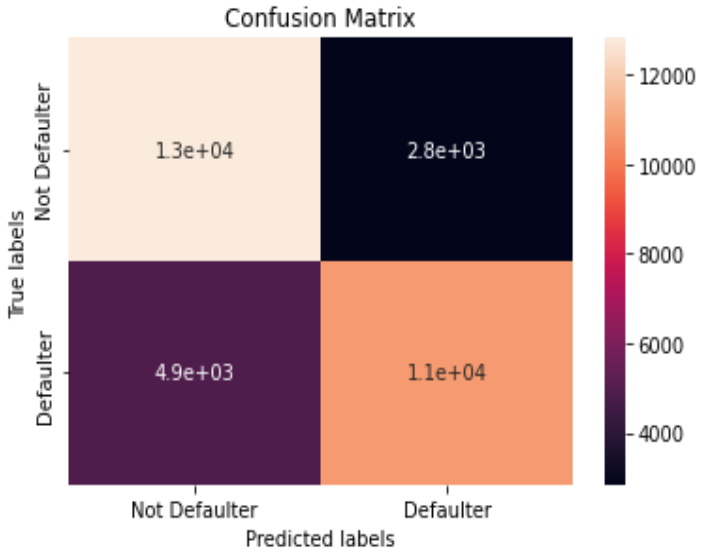
According to the actual situation, for unbalanced data, we should use the evaluation index of unbalanced data [21], but because at the beginning of the experiment, we have balanced the number of positive and negative classes in the sample. And we are still using the two-class evaluation indicators commonly used in the past: hybrid matrix, recall, precision, f1-score, AUC value, and so on.



approx. 73%. As we have imbalanced dataset, F1- score is better parameter. Let's go ahead with other models and see if they can yield better result.

8. Implementing Logistic Regression

Logistic Regression is one of the simplest algorithms which estimates the relationship between one dependent binary variable and independent variables, computing the probability of occurrence of an event. The regulation parameter C controls the trade-off between increasing complexity (overfitting) and keeping the model simple (underfitting). For large values of C, the power of regulation is reduced and the model increases its complexity, thus overfitting the data.



9. DECISION TREE:

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like

We have implemented logistic regression and we getting f1-sore

tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

10. RANDOM FOREST:

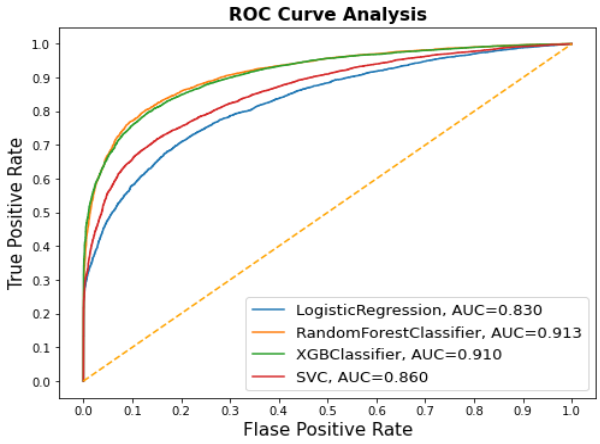
Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the greatest number of times a label has been predicted out of all.

11. GRADIENT BOOSTING:

The term gradient boosting consists of two sub-terms, gradient and boosting. We already know that gradient boosting is a boosting technique. Let us see how the term 'gradient' is related here. Gradient boosting re-defines boosting as a numerical optimization problem where the objective is to minimize the loss function of the model by adding weak learners using gradient descent. Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc.

12. ROC Curve

The area under the ROC curve tells us how well the model separates the different classes in the dataset. It plots true positive rate against false positive rate



13. Conclusion

XGBoost model has the highest recall, if the business cares recall the most, then this model is the best candidate. If the balance of recall and precision is the most important metric, then Random Forest is the ideal model. Since Random Forest has slightly lower recall but much higher precision than Logistic Regression, I would recommend Random Forest.

- Data categorical variables had minority classes which were added to their closest majority class
- There was not huge gap but female clients tended to default the most.
- Labels of the data were imbalanced and had a significant difference.
- Gradient boost gave the highest accuracy of 82% on test dataset.

14. REFERENCES:

- https://book.akij.net/eBooks/2018/May/5aef50939a868/Data Science_ for_Bus.pdf
- Hands-On Exploratory Data Analysis with Python Perform EDA techniques to understand, summarize, and investigate your data by Suresh Kumar Mukhiya, Usman Ahmed (z-lib.org)
- https://bunker2.zlibcdn.com/dtokend/01c5fc197a94283bfb0c0943bd5b2d0c

- The best **accuracy** is obtained for the **Random Forest** and **XGBoost classifier.**
- Furthermore, ** accuracy** does not consider the rate of **false positives** (non-default credits cards that were predicted as default) and **false negatives** (default credit cards that were incorrectly predicted as non-default).

Both cases have negative impact on the bank, since **false positives** leads to unsatisfied customers and **false negatives** leads to financial loss.

- From above table we can see that X**GBoost Classifier** having **Recall, F1-score**, and **ROC Score** values equal 82%, 77%, and 86% and **Random Forest Classifier** having **Recall, F1-score**, and **ROC Score** values equal 81%, 75%, and 84%.
- **XGBoost Classifier** and **Decision Tree Classifier** are giving us the best **Recall, F1-score**, and **ROC Score** among other algorithms. We can conclude that these two algorithms are the best to predict whether the credit card is default or not default according to our analysis.
- The best **accuracy** is obtained for the **Random Forest** and **XGBoost classifier.**
- Furthermore, ** accuracy** does not consider the rate of **false positives** (non-default credits cards that were predicted as default) and **false negatives** (default credit cards that were incorrectly predicted as non-default).

Both cases have negative impact on the bank, since **false positives** leads to unsatisfied customers and **false negatives** leads to financial loss.