Reunion Assessment

Shubham Naik

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 and SMOTE. 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 for prediction.

The model effectively solves the problem of sample data imbalance. At the same time, this paper constructs common machine learning models, logistics, SVM, random forest, and tree, and compares the classification performance of these prediction models.

Problem Statement

This project is aimed at predicting whether an applicant who is about take loan is a risky applicant or not.

Understanding the Data

This dataset contains information on high-risk applicants.

This data tells us about the applicants that were flagged as high risk and were not allowed to take any loan.

Attributes-

- **Applicant_id:** Contains ID for each applicant.
- **Primary_applicant_age_in_years:** Age of the applicant in years.
- **Gender:** Applicant's gender.
- **Marital_status:** Tells us about the marital status of the applicant.
- **Housing:** Type of housing the applicant resides at.
- **Years_at_current_residency:** Years an applicant has spent in a particular residency.
- **Employment_status:** Applicant employment status.
- **Empd_for_atleast:** Tells us about the number of days the applicant was employed.
- **Has_been_employed_for_at_most:** Maximum number of days the applicant was employed.
- **Telephone:** Applicants telephone number
- **Foreign_worker:** Tells us that the applicant is foreign worker or not.
- **Savings_account_balance:** Tells us about the savings account balance of applicant.
- **Months_loan_taken_for:** Tells us about how many months the applicant has taken the loan.

- **Purpose:** Purpose of applicant the loan was taken for.
- **Principal_loan_amount:** Amount of loan that was taken by the applicant.
- **EMI_rate_prcnt:** EMI rates in percentage for each applicant.
- **Property:** Tells us about the property owned by the applicant.
- **Other_EMI_plans:** Tells us about the other EMI plans the applicant has.
- **Number_of_existing_loan_at_this_bank:** Tells us about the other loans the applicant has in this bank.
- **Loan_history:** Applicants loan history records.
- **High_risk_applicant:** Tells us that the applicant has been flagged as a high risk or not.

Exploratory Data Analysis

The main goal of EDA is to understand the data and represent in such a way that everything given in the data makes sense. So far, we've got to know about which data has been given to us and also, we've found no missing values in it. So now what? The answer is quite simple, we now analyze what are the key information we can gather from the given data, which not only helps us to better understand about the data but also it is quite simple for any non-technical person to understand it too. For now, we will look at the EDA part.
With some quick analysis we found that there are more than 6 features that are highly correlated with each other. There are a greater number of low credit risk applicants than high credit risk applicants. Over 69% of applicants are male whereas the remaining 31% are female. There is more applicant with marital status as single than married or divorced. Many applicants have their own housing property and their marital status is single. Over 400 applicants have been residing in the same residency for 4 years. Among all applicants 629 are skilled employee whereas 199 are unskilled. The main purpose for getting the loan is to buy electronic equipment or new vehicle. Over 620 applicants already have pending loan in the same bank and 693 of such applicants have been flagged as high-risk applicants.

Data Pre-processing

Handling missing values
Missing values were handled by filling the 'NaN' field from the 'empd_for_atleast' column to '0 year'. Same was done for the 'has_been_employeed_for_at_most' column which was replaced with '7+ year'.
To fill the missing values in the housing column some conditions were applied, like if the housing was own then it would be assigned as real estate or if the housing is rent then it would be car or other and for free housing it was assumed to be building society savings agreement/life insurance.

SMOTE
We know that smote is a method for synthesizing new samples and solving data imbalance 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. This algorithm can reduce the imbalance between categories on the one hand and reduce the imbalance within categories on the other hand.

- Original dataset shape 988.
- Resampled dataset shape 1386.

Algorithms

Now our data is ready to be used by our models for training. Let's train few models and compare their scores.

Logistic Regression

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. Logistic regression is applied to predict the categorical dependent variable. In other words, it's used when the prediction is categorical, for example, yes or no, true or false, 0 or 1. The predicted probability or output of logistic regression can be either one of them, and there's no middle ground. For this model the penalty were set as l1 and l2 along with 'C: 0.001,0.01,0.1,1,10,100,1000' After which the best parameters were 'C:0.1, penalty:l2'. The best score was 0.716. The train data result was 0.725 and test data result were 0.689. The final F1 score was 0.69 along with ROC_score of 0.689.

SVC Model

Support Vector Classifier, is a supervised machine learning algorithm typically used for classification tasks. SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes. The parameters were set as 'C' 0.1,1,10,100 with 'kernel' as 'rbf'. A StandardScaler was used to transform the data for the model. After using the GridSearchCV the best parameters were 'C' 10 with 'kernel' as 'rbf' having best score of 0.753. The overall accuracy on train data was 0.73 and on test data it was 0.69. The final F1 score was 0.718 with ROC_score of 0.723.

Decision Tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Once the model was fit with the data, we then found the best parameters for our model, which were, 'max_depth: 20,30,50,100' and 'min_samples_split: 0.1,0.2,0.4'. The test accuracy score was 0.64 and train accuracy was 0.73, which is almost same when we compare it with SVC. The final F1 score was 0.65 with ROC_score of 0.6444.

Random Forest

Random Forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Once the model was fit with the data, we then found the best parameters for our model, which were, 'max_depth: 10,20,30' and 'n_estimators: 100,150,200'. The test accuracy score was 0.75 and train accuracy was 1.0. We can see from above results that we are getting around 100% train accuracy and 75% for test accuracy which depicts that model is overfitting. However, our f1-score is around 74%, which is not bad. We then plot a graph showing the features with most weightage given by our model and found out that attributes like 'Principal_loan_amount', 'Months_loan_taken_for' and

'Primary_applicant_age_in_years' had the highest importance amongst others.

XGBoost Model

XGBoost is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples. Once the model was fit with the data, we then found that the train data had score of 0.93 whereas the test data had score of 0.733. And finally, the F1 score was 0.733 with ROC_score of 0.7331

Hyperparameter Tuning

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned. We used the Hyperparameter tuning on the XGBoost and then we retrained the model, after which it was found out that it had train data score of 0.995 and test data score of 0.75. And finally, the F1 score was 0.753 with ROC_score of 0.7502.

Conclusion

- XGBoost provided us the best results giving us a recall of 76 percent (meaning out of 100 risk applicants 76 will be having high chances of paying loan back)
- Random Forest also had good score as well.
- Logistic regression being the least accurate with a recall of 69.