**Credit Card Default Prediction**

**Shubham Naik**

**Data science trainee,**

**Almabetter, Bangalore**

**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.

***Keywords: machine learning,* *default payment next month, limit balance, bill amount, payment amount***

1. **Problem Statement**

### This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the [K-S chart](https://www.listendata.com/2019/07/KS-Statistics-Python.html) to evaluate which customers will default on their credit card payments

1. **Understanding the Data**

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients Taiwan from April 2005 to September 2005.

Credit card default happens when you have become severely delinquent on your credit card payments. Default is a serious credit card status that affects not only you’re standing with that credit card issuer but also your credit standing in general and your ability to get approved for other credit-based services.   
  
**Attributes-**

* **LIMIT\_BAL:** Amount of the given credit it includes both the individual consumer credit  
  and his/her family credit.
* **SEX:** Gender (1 = male; 2 = female).
* **EDUCATION:** Education (1 = graduate school; 2 = university; 3 = high school; 4 =  
  others).
* **MARRIAGE:** Marital status (1 = married; 2 = single; 3 = others).
* **AGE:** Age (year).
* **PAY\_0 - PAY\_6:** History of past payment from April to September 2005.
* **BILL\_AMT1 - BILL\_AMT6:** Amount of bill statement from April to September 2005.
* **PAY\_AMT1 - PAY\_AMT6:** Amount of previous payment from April to September 2005.
* **Default payment next month:** default payment (Yes = 1, No = 0), as the response  
  variable.

1. **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 almost 22% of clients were ‘default next month’ and most of clients had Education level of graduate school and university. Average age of clients was ~36 years and finally the average value for the amount of credit card limit was 167,484, implying that most of defaults are for credit limits 0-100,000. Gender was no barrier; data was evenly distributed amongst males and females. With the help of correlation matrix, we found that correlation is decreasing with distance between months, lowest correlations were between Sept-Apr.

We also used Boxplots which stated that, Mean values are generally smaller for males than for females, with few exceptions, for example at age 39, 48, until approximately 60, where mean values for males are generally larger than for females.

1. **Data Pre-processing**

* **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 30000.

● Resampled dataset shape 46728.

* **Feature Engineering**

The default prediction model based on machine learning requires extracting features of the transaction flow data. The quality of the model largely depends on the application of feature engineering

1. **Algorithms**

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

1. **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. Once the model was fit with the data, we then found the best parameters for that model which were ‘C: 1000’ and ‘penalty: L2’. The test accuracy score of this model was 0.752 and it performed the same on train data too. We then plotted a confusion matrix for both train and test with labels as ‘Not Defaulter’ and ‘Defaulter’. We then plot a graph showing the features with most weightage given by our model and found out that attributes like ‘PAY\_6\_1’, ‘PAY\_5\_1’ and ‘EDUCATION\_1’ had the highest importance amongst others.



1. **SVM Model**

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving SVM model sets of labeled training data for each category, they're able to categorize new text. Once the model was fit with the data, we then found the best parameters for our model, which were, ‘C: 10’ and ‘kernel: rbf’. The test accuracy score was 0.752 and train accuracy was 0.752 as well. But when we found the f1-score it was approx. 76%, which implies that there was more room for improvement. We then plotted a confusion matrix for both train and test with labels as ‘Not Defaulter’ and ‘Defaulter’.



1. **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’ and ‘min\_samples\_split: 0.1’. The test accuracy score was 0.70 and train accuracy was 0.70, which is bad when we compare it with SVM or Logistic models.



1. **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: 30’ and ‘n\_estimators: 150’. The test accuracy score was 0.83 and train accuracy was 0.99. We can see from above results that we are getting around 99% train accuracy and 83% for test accuracy which depicts that model is overfitting. However, our f1-score is around 82%, 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 ‘LIMIT\_BAL’, ‘PAY\_AMT1’ and ‘BILL\_AMT1’ had the highest importance amongst others.



1. **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](https://en.wikipedia.org/wiki/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 the best parameters for our model, which were, ‘max\_depth: 10’ and ‘min\_child\_weight: 6’. The test accuracy score was 0.77 and train accuracy was 0.78. The f1-score was 0.75 which is not bad at all.

1. **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.

1. **Conclusion**

We started with LogisticRegression, for which we obtained an AUC score of 0.829.  
Next we used a RandomForestClassifier, for which we obtained an AUC score of 0.912. Then we experimented with an XGBoost model, for which the AUC score obtained was 0.910. We then used SVC the obtained values of AUC for the SVC were around 0.858.

**References-**

1. GeeksforGeeks
2. Analytics Vidhya
3. Hindawi
4. Towards data science
5. Medium