**Credit Card Default Prediction**

**Sharath Diwakar, Mohd. Danish, Abdul Rahman Talha**

**Data Science Enthusiasts**

**AlmaBetter, Bengaluru**

**Abstract**

Credit risk plays an important role in the banking industry business. Credit card has been one of the most booming businesses in banking sectors but with its pros comes some cons also. With the growing numbers of credit card users, the default rate is also increasing which is jeopardizing the banking sector. So, to tackle this growing problem the world needs solutions and one of it is to try and predict in time which customer can result in a default.

In this report we are applying various Machine Learning algorithms to predict and analyze the default rate and hence to see which all features are important to make a robust and efficient model.

***Keywords: Credit Card, Default Payment, Classification Models***

1. **Problem Statement**

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.

To build a robust and accurate Machine Learning model which can help the banks in predicting Credit Card Default by classifying defaulters and non-defaulters.

1. **Introduction**

The dataset which contains data of clients from a financial institute in Taiwan is provided by UCI Machine Learning Repository. We will study this data to predict the client’s chances to default on his/her credit card. High credit card default rates can make a business in trouble even bankrupt. To default is to fail to make a payment on a debt by the due date. The health of the credit card industry is best measured not by the number of people with cards, but rather the number who pay their bills.

Serious delinquency rates are measured as the percentage of balances that are 90 or more days past the due date. The delinquency rate indicates the percentage of past-due loans within the borrowers’ entire loan portfolio.

In a developing country like ours this plays an important role as there are many entrepreneurs coming up and they need money to start their business which is funded by loans and more money is needed by them to set up their business so default rate also needs to be in check by the banks or else they can run into serious problems.

Hence a need for a risk prediction model comes into existence which should be able to classify the probability of defaulters and non-defaulters.

Credit card default prediction is one of the main predictions that banks are concerned with including credit scoring to better understand why customers are likely to default.

1. **Dataset Insight**

The dataset was taken from UCI Machine Learning Repository. The dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

**Column Information**

1. **ID**: ID of each client

2. **LIMIT\_BAL**: Amount of given credit in NT dollars (includes individual and

family/supplementary credit

3. **SEX**: Gender (1=male, 2=female)

4.**EDUCATION**:(1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

5. **MARRIAGE**: Marital status (1=married, 2=single, 3=others)

6. **AGE**: Age in years

7. **PAY\_0**: Repayment status in September, 2005 (-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)

8. **PAY\_2**: Repayment status in August, 2005 (scale same as above)

9. **PAY\_3:** Repayment status in July, 2005 (scale same as above)

10. **PAY\_4**: Repayment status in June, 2005 (scale same as above)

11. **PAY\_5**: Repayment status in May, 2005 (scale same as above)

12. **PAY\_6**: Repayment status in April, 2005 (scale same as above)

13. **BILL\_AMT1**: Amount of bill statement in September, 2005 (NT dollar)

14. **BILL\_AMT2**: Amount of bill statement in August, 2005 (NT dollar)

15. **BILL\_AMT3**: Amount of bill statement in July, 2005 (NT dollar)

16. **BILL\_AMT4**: Amount of bill statement in June, 2005 (NT dollar)

17. **BILL\_AMT5**: Amount of bill statement in May, 2005 (NT dollar)

18. **BILL\_AMT6**: Amount of bill statement in April, 2005 (NT dollar)

19. **PAY\_AMT1**: Amount of previous payment in September, 2005 (NT dollar)

20. **PAY\_AMT2**: Amount of previous payment in August, 2005 (NT dollar)

21. **PAY\_AMT3**: Amount of previous payment in July, 2005 (NT dollar)

22. **PAY\_AMT4**: Amount of previous payment in June, 2005 (NT dollar)

23. **PAY\_AMT5**: Amount of previous payment in May, 2005 (NT dollar)

24. **PAY\_AMT6**: Amount of previous payment in April, 2005 (NT dollar)

25. **default. payment. next. month**: Default payment (1=yes, 0=no)

**6. Steps involved:**

* **Exploratory Data Analysis**

After loading the dataset, we compared our target variable that is “Default payment next month” with other independent variables. This process helped us figure out various aspects and relationships among the dependent and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the dependent variable.

* **Null values Treatment**

No presence of null values in the dataset.

* **Encoding of categorical columns**

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **Random Forest Classifier**
3. **XGBoost Classifier**

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree-based models

like Random Forest Classifier and XGBoost classifier.

**7.1. Algorithms:**

1. **Logistic Regression:**

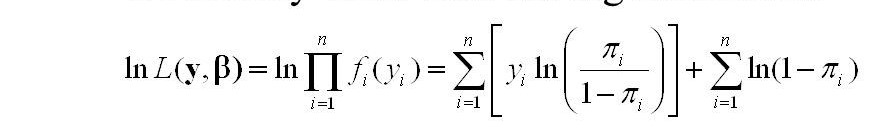
Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:



1. **Random Forest Classifier:**

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 most number of times a label has been predicted out of all.



1. **XGBoost**

To understand XGBoost we have to know gradient boosting beforehand.

* **Gradient Boosting**

Gradient boosted trees consider the

special case where the simple model

is a decision tree

****

In this case, there are going to be 2

kinds of parameters P: the weights at

each leaf, w, and the number of

leaves T in each tree (so that in the

above example, T=3 and

w=[2, 0.1, -1]).

When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

**XGBoost** is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

**7.2. Model performance:**

Model can be evaluated by various metrics such as:

**1. Confusion Matrix**

The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

1. **Precision/Recall**

**Precision** **is the** ratio of correct positive predictions to the overall number of positive predictions: **TP/TP+FP.**

**Recall is the** ratio of correct positive predictions to the overall number of positive examples in the set: **TP/FN+TP.**

1. **Accuracy**

**Accuracy is** given by the number of correctly classified examples divided by the total number

of classified examples. In terms of the confusion matrix, it is given by: **TP+TN/TP+TN+FP+FN.**

1. **Area under ROC Curve (AUC)**

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

**7.3. Hyper parameter tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV for hyperparameter tuning. This also results in cross validation and in our case, we divided the dataset into different folds..

1. **Grid Search CV:** Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

**8. Conclusion:**

Data exploration, visualization, and relationship between different features was carried out and then investigated two predictive models. The data was split into three parts i.e., training set, validation set and testing set.

We started with Logistic Classifier and then with RandomForestClassifier, for which we obtained an AUC code of 0.81 and 0.91, respectively, when predicting the target for the test set.

We then used the XGBoost model, validation score 0.912 was obtained. Then we used the model with the best training step to predict the target value from the test data; the AUC score obtained was 0.910 which is less than the validation score of random forest.

***Random forest performed best in this problem data set with highest recall more than 86% and high precision for random forest was obtained and highest AUC score of around 0.912.***

With the growing number of credit card users, banks have been facing an escalating

credit card default rate this model can help the banks to classify credit card default by making better decisions like which features are important when the bank needs to issue a credit card or what should be the credit limit for a particular person. Banks in such a way can make the most of the machine learning models which can contribute in boosting their performance and image in the industry.

**References**

1. Stack Overflow
2. Medium
3. GeeksforGeeks