Korea Advanced Institute of Science and Technology

Introduction to Financial Engineering

Clustering Bank Customers and Predicting Their Loan Status

Professor:
Woo Chang Kim

Student:
Federico Berto

Course ID:
IE471

ID number:
20204817

Introduction

The goal of this report is to describe the experimental process and result for predicting bank loan statuses of customers based on a variety of collected data.

In particular, we will use *K-means clustering* and *PCA analysis* for visualizing the clusters. As for the AI algorithms, below is a summary of each.

AI algorithms

XGBoost

XGBoost [1] is an open-source and scalable library employing regularized gradient boosting. It has become famous for winning many machine learning competitions.

CatBoost

CatBoost [2] is a gradient boosting algorithm which can handle categorical values as well compared to XGBoost, by using *one-hot coding*.

Light GBM

Light GBM [5] improves over CatBoost's categorical feature hot-coding by using a special algorithm finding the split value of categorical features, Gradient-based One-Side Sampling (GOSS).

Multilayer Perceptron Classification with Imbalanced Datasets

Multilayer Perceptrons with Imbalanced Datasets [4] are very different in terms of structure compared to the previous algorithms. We use Deep Learning and Backpropagation for obtaining the optimal weights minimizing a loss function. Moreover, weighting and other techniques improve over the dataset imbalance.

Explainable AI

In order to provide a better visualization to the results, we use Explainable AI (XAI). In particular we will employ these two algorithms:

1. **LIME**: Local Interpretable Model-Agnostic Explanations [7] , this learns a locally interpretable prediction model for any classifier.
2. **SHAP**: SHapley Additive exPlanations [6], this algorithm used class importance features for showing human-understandable machine learning.

Experiments

Clustering and prediction for Kaggle's Bank Loan Status Dataset

In this section we provide the required data for the report based on the Kaggle's Bank Loan Status Dataset [1].

1. K-means cluster plot with the PCA results (Figure 1)

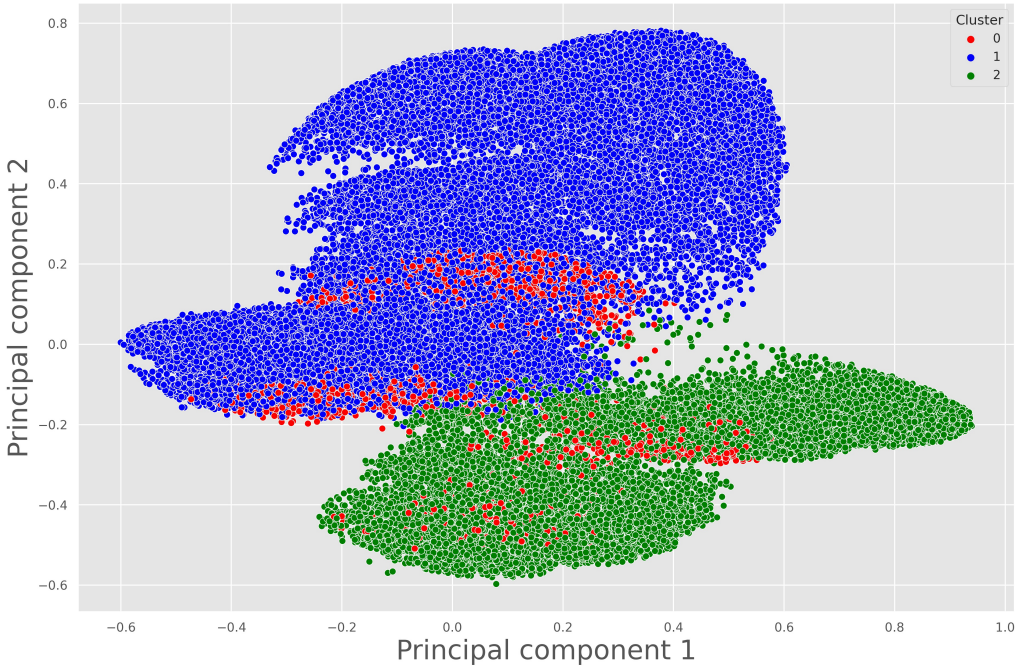

Figure 1: PCA results on the clustering

2. Comparison of the loan status proportion of those three clusters:

Loan Status	Mean	Cluster 0	Cluster 1	Cluster 2
Charged Off	22.6336	19.952	19.9007	30.3494
Fully Paid	77.3664	80.048	80.0993	69.6506

As we can see from this analysis, the K-means algorithm grouped the customers in three main clusters: while the first two (Clusters 0 and 1) have roughly an 80-20 ratio of people who managed to pay the loan and people who were charged off,

[1] Source: Kaggle

Cluster 2 has a much higher ratio of people who could not pay off the loan (more than 30%). Therefore, it is more probable for people in this cluster to *default* the payment and be untrustworthy to the bank.

3. Columns, non-null count in those columns, and data type of original data using `credit.info()`:

Feature No.	Value	Number	Type
0	Loan ID	100000 non-null	object
1	Customer ID	100000 non-null	object
2	Loan Status	100000 non-null	object
3	Current Loan Amount	100000 non-null	float64
4	Term	100000 non-null	object
5	Credit Score	80846 non-null	float64
6	Annual Income	80846 non-null	float64
7	Years in current job	95778 non-null	object
8	Home Ownership	100000 non-null	object
9	Purpose	100000 non-null	object
10	Monthly Debt	100000 non-null	float64
11	Years of Credit History	100000 non-null	float64
12	Months since last delinquent	46859 non-null	float64
13	Number of Open Accounts	100000 non-null	float64
14	Number of Credit Problems	100000 non-null	float64
15	Current Credit Balance	100000 non-null	float64
16	Maximum Open Credit	99998 non-null	float64
17	Bankruptcies	99796 non-null	float64
18	Tax Liens	99990 non-null	float64

4. Count plot of the *Years in current job* feature:

5. Shapley value of 14th datum is 1.97 and the analysis result is as following:

6. Summary of all feature effects and description:

Figures 3 and 4 show the importance of the features based on the SHAP values, while Figure 6 shows the features importance based on LIME, another Explainable AI algorithm.

As we can see from the analysis, the most import three features are the Current Loan Amount, Credit Score and the Annual Income. In particular, the SHAP values show that high values of Current Loan Amount and Annual Income are positively

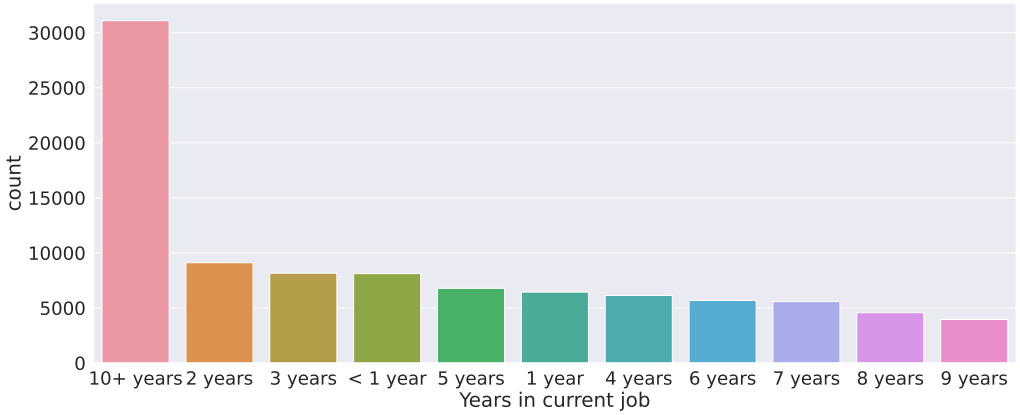

Figure 2: Count plot of years in the current job

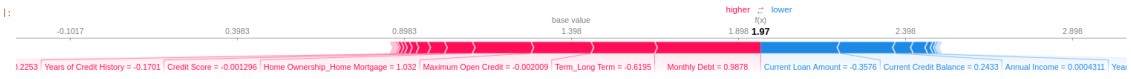

Figure 3: Force plot of the 14th datum

correlated to the prediction, while high values of credit score are negatively correlated to it.

Analyzing and improving the Predictions

AI algorithms

In this section, we will analyze the prediction results on the test dataset by comparing `XGBoost` to other boosting algorithms: `CatBoost` and `LightGBM`.

Moreover, we also implement a version of Multilayer Perceptrons with Pytorch Lightning [3] and and *Imbalanced Dataset Sampler* to compare the results of Deep Learning and Boosting Algorithms.

In order to compare the results, we will use *confusion matrices*, which are a simple way of visualizing classification results. Moreover, we will not only employ Mean Squared Error, but also *precision* and *recall* measures to have a better understanding of how the algorithms perform.

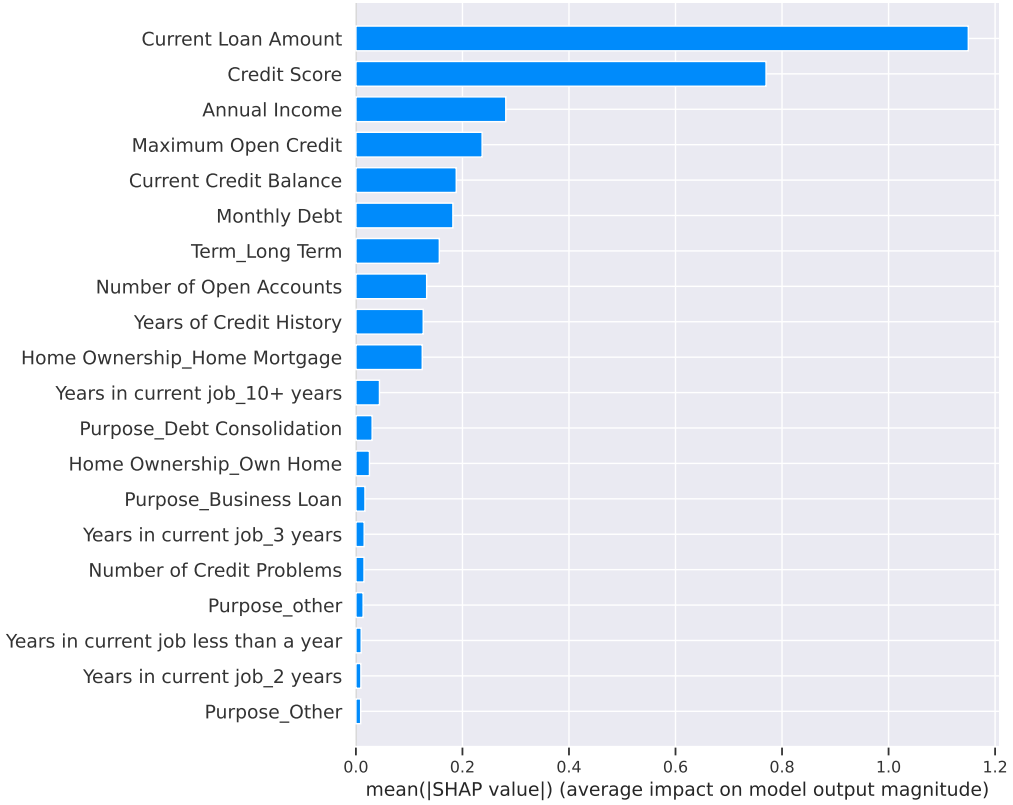

Figure 4: Bar summary plot with SHAP

Results

Results Comparison

We report the table of precision, recall and F-score for the rejected loans and MSE for each algorithm on the training dataset:

Comparison of metrics on the test dataset				
	XGBoost	**CatBoost**	**Light GBM**	**MLP**
Precision	0.7611	**0.8916**	0.4136	0.3785
Recall	0.2650	0.2335	**0.6302**	0.5468
F-score	0.3931	0.3701378	**0.4994**	0.4473
MSE	0.1862	0.1809	**0.1797**	0.3109

The results show that LightGBM could achieve a lower MSE error and also beat the others for Recall and F-score on the class of Unpaid Loans. The MLP could not beat the competition, however we have to notice that the competing models are state of the art for classification on similar datasets. Classifications on others such

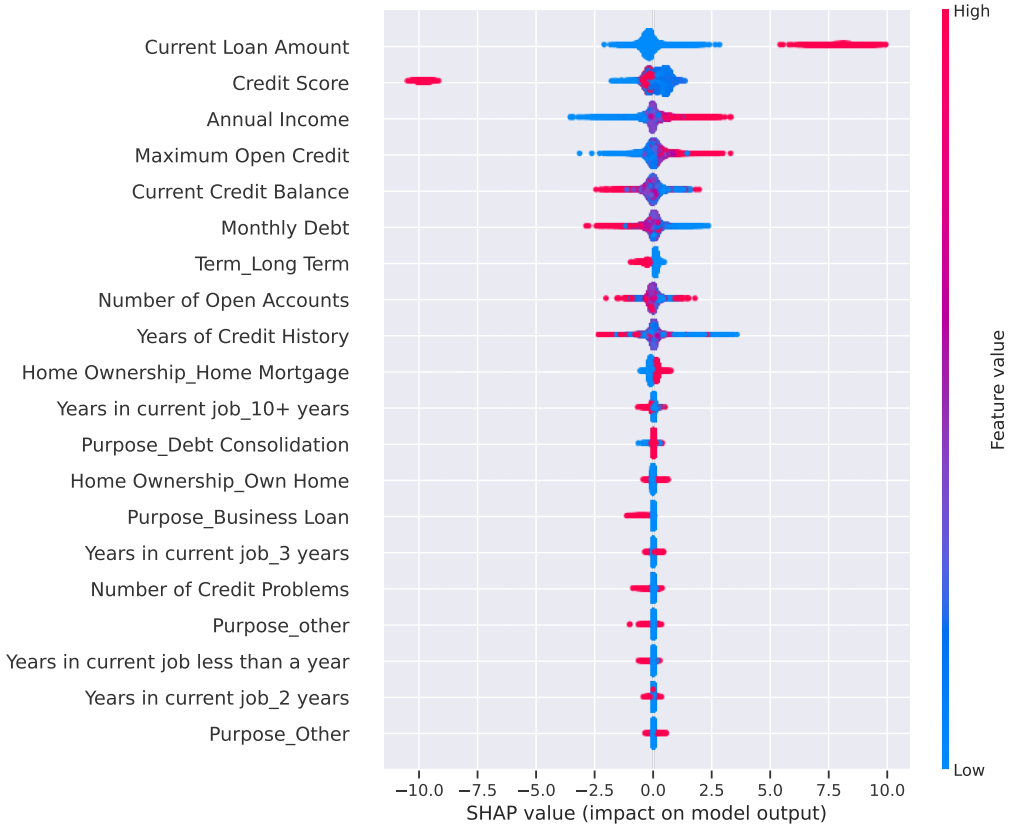

Figure 5: Tree summary plot with SHAP

Figure 6: Summary with LIME showing "good" and "bad" feature depending on our prediction labels

as images would have probably achieved different results.

Even though the precision of LightGBM is lower than i.e., XGBoost and CatBoost, we can see how LightGBM shines in a much higher Recall and F-score. In practice,

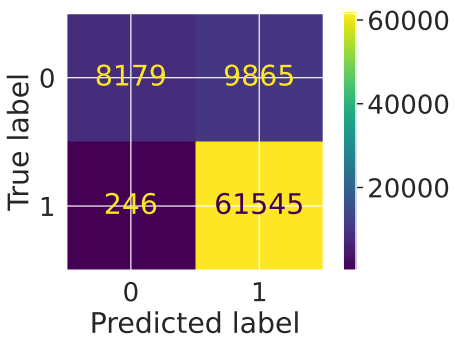

Figure 7: Train dataset confusion matrix for XGBoost

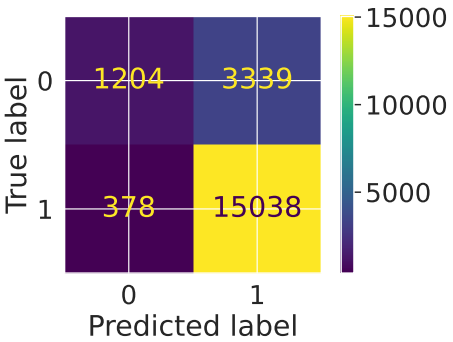

Figure 8: Test dataset confusion matrix for XGBoost

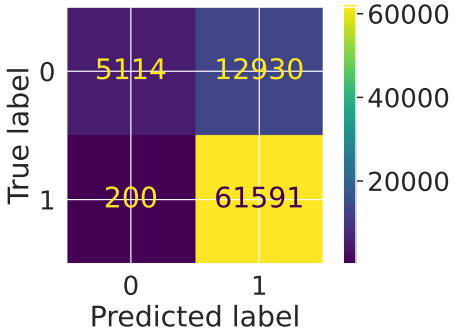

Figure 9: Train dataset confusion matrix for CatBoost

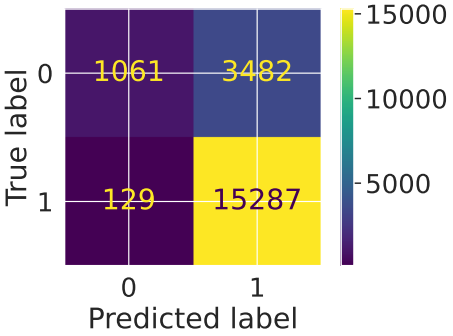

Figure 10: Test dataset confusion matrix for CatBoost

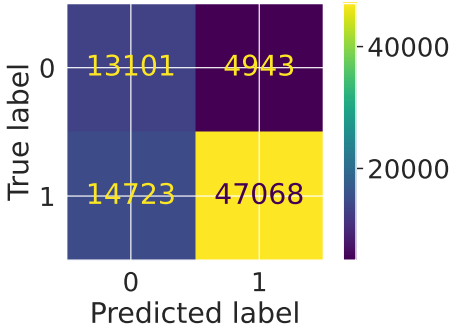

Figure 11: Train dataset confusion matrix for Light GBM

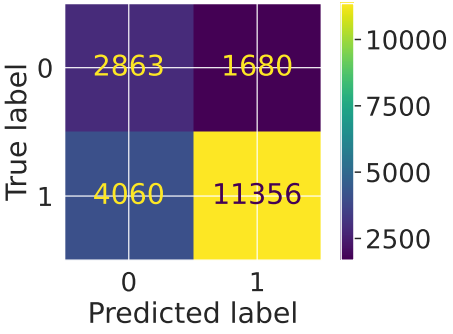

Figure 12: Test dataset confusion matrix for Light GBM

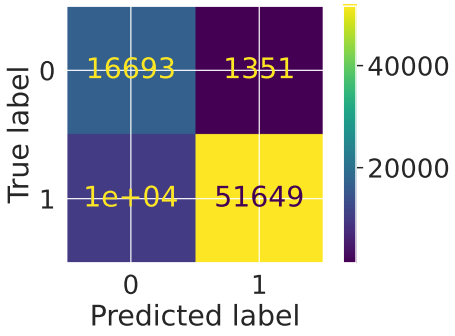

Figure 13: Train dataset confusion matrix for the Multilayer Perceptron

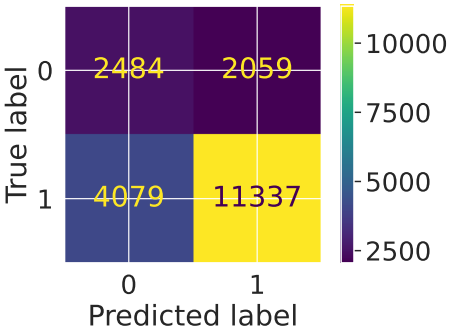

Figure 14: Test dataset confusion matrix for the Multilayer Perceptron

the Recall becomes higher when the number of False Negatives is lower i.e., the number or loan statuses predicted as "Fully paid" incorrectly. A bank would be probably more interested in having a higher number of Recall, because the number of actual prediction for people going to default loans will be higher (see the Figures about confusion matrices).

Even though not being too "precise", it could be more important to detect more possible "bad" customers with a higher recall and perhaps being more careful about that cluster. In this respect, even the Multilayer Perceptron turned out to be better than the former two Boosting Algorithms.

Clustering and prediction for UCI Credit Card Dataset

In this section, we use another dataset, namely the UCI Credit Card Dataset [2]. Information about the features are available at the link.

This dataset has as target the prediction of *defaulting* probability in the next month i.e., not being able to pay, which is labeled as 1. In other words, the results are switched when comparing the inability to pay the next month, but we apply the same type of analysis.

Loan Status	Cluster 0	Cluster 1	Cluster 2
Payment	81.1947	83.5468	73.3544
Payment Default	18.8053	16.4532	26.6456

As we can see, Cluster 2 has a much higher percentage of defaulting customers when compared to others.

[2]Source: UCI

Figure 15: PCA results on the clustering

The results show that `PAY_0` is the most important feature for determining the model's result: the feature indicates if the customer has repaid the loan in September 2005. `LIMIT_BAL` indicates the amount of credit and `BILL_AMT1` is the amount of bill statement. Interestingly, the credit score is the second most important feature for both datasets.

Comparison of metrics on the test dataset				
	XGBoost	**CatBoost**	**Light GBM**	**MLP**
Precision	0.5968	**0.6667**	0.4535	0.4085
Recall	0.3663	0.3625	**0.6275**	0.5065
F-score	0.4540	0.4697	**0.5265**	0.4522
MSE	0.1928	0.1792	**0.1742**	0.2940

We consider Precision, Recall, F-score for the test data of defaulting customers and the Mean Squared Error metric. We can see how the previous results match these: Light GBM is able to obtain better results with Recall, F-score and the MSE metrics.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge*

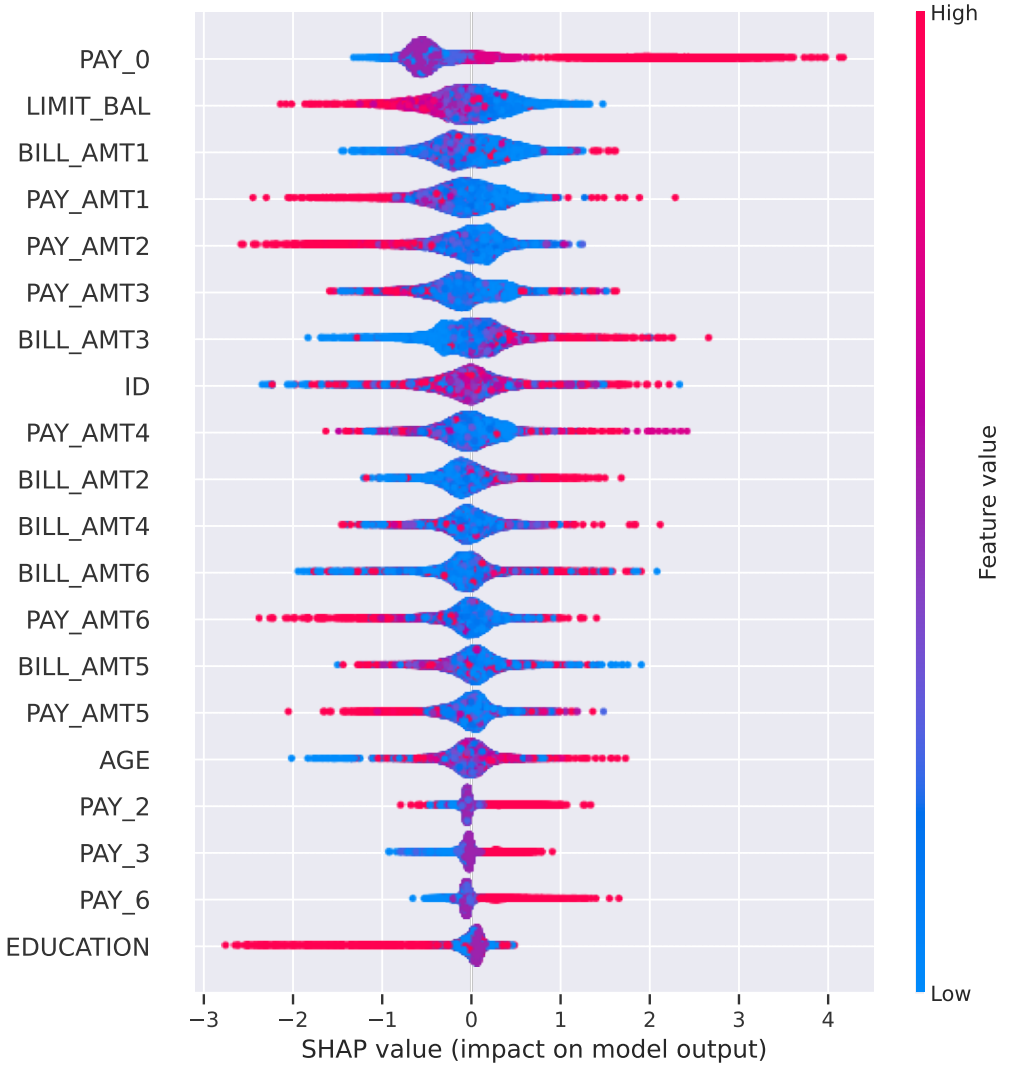

Figure 16: Summary SHAP plot for training data and XGBoost predictions

discovery and data mining, pages 785–794, 2016.

[2] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[3] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[4] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Trans-*

actions on knowledge and data engineering, 21(9):1263–1284, 2009.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[6] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017.

[7] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.