Model selection:

Because we directly developed the model using the imbalanced data, we cannot simply use accuracy to evaluate and compare models directly. We use f1 score, AUC, combined with accuracy, and recall. Here are the results

Table 1. Hyperparameter optimization and modeling results using the original data

Algorithms	Hyperparameters	best estimate	accuracy	precision	recall	f1 score	AUC	runtime	top 3 most important features
regularized logistic regression	C	50	0.809	0.68	0.234	0.348	0.725	4.77	PAY_1, BILL_AMT1, BILL_AMT2
decision tree	criterion, max_depth	gini, 4	0.819	0.66	0.349	0.457	0.725	36.73	PAY_1, PAY_2, PAY_AMT3
random forest	n_estimators, max_depth	200, 9	0.821	0.67	0.356	0.465	0.774	85.31	PAY_1, PAY_2, PAY_3
gradient boosting	learning_rate, n_estimator, max_depth	50, 2, 0.1	0.822	0.68	0.349	0.46	0.771	1926.3	PAY_1, PAY_2, BILL_AMT1
Extreme gradient boosting	subsamples, n_estimators, max_depths, learning rate, gamma, reg_alpha	0.5, 200, 5, 0.05, 0.1, 0	0.819	0.66	0.358	0.463	0.778	301.9	PAY_1, PAY_1, PAY_3
KNN	n_neighbors, p	50, 2	0.808	0.646	0.258	0.368	0.741	100.8	PAY_1, PAY_2, PAY_3

Table 2. Hyperparameter optimization and modelling results using the engineered features

Algorithms	Hyper-parameters	best estimate	accuracy	precision	recall	f1 score	AUC	top 3 most important features
regularized logistic regression	C	50	0.797	0.672	0.176	0.279	0.692	pay_sum, payment_sum, LIMIT_BAL
decision tree	criterion, max_depth	entropy, 4	0.801	0.625	0.276	0.383	0.743	pay_sum, payment_sum, bill_sum
random forest	n_estimators, max_depth	200, 9	0.801	0.612	0.299	0.398	0.762	pay_sum, payment_sum, bill_trend
gradient boosting	learning_rate, n_estimator, max_depth	50, 2, 0.1	0.802	0.616	0.299	0.402	0.759	pay_sum, payment_sum, bill_sum
Extreme gradient boosting	subsamples, reg_alpha, n_estimators, max_depths, learning rate, gamma	0.7, 0.05, 400, 1, 0.1,0.4	0.803	0.614	0.322	0.422	0.759	pay_sum, payment_sum, bill_sum
KNN	n_neighbors, p	2, 50	0.796	0.636	0.204	0.309	0.713	pay_sum, payment_sum

Based on f1score and AUC, we can see random forest and XGB are the best choice when we choose the original data set to develop the model. XGB is the highest performer when we use the reduced data set, df_sum. Their accuracy and recall scores are also high compared with the other algorithms.