

SmartCampus: Digital Financial Innovation Proposal for Hong Kong Universities



Jiang Feiyu [3035770800]

HKU FITE7410 Financial fraud analytics

April 16 2025

Overview

This study explores the application of machine learning techniques for detecting financial fraud transactions using a comprehensive dataset of 100,000 transactions. Different data analytics techniques, such as Random Forest and Logistic Regression models, are utilized in the analysis to create a reliable fraud detection system.

1. Background and Objectives of the Case Study

Financial fraud poses a persistent and evolving threat to the global banking system, resulting in significant monetary losses and eroding trust in financial institutions. As digital transactions continue to grow exponentially, traditional manual fraud detection methods have proven increasingly inadequate in addressing sophisticated fraudulent activities. The emergence of machine learning technologies offers a promising solution for developing automated, real-time fraud detection systems capable of processing large volumes of transactions while maintaining high accuracy (West & Bhattacharya, 2016).

The study utilizes a real-world dataset comprising 100,000 financial transactions, each characterized by 101 distinct features. The preliminary examination indicates a significant disparity in class distribution within the dataset, with 88,682 valid transactions (88.68%) and 11,318 unauthorized transactions (11.32%). This imbalance reflects the typical distribution observed in real-world fraud scenarios, where fraudulent activities represent a minority of total transactions.

The main goal of this study is to create and assess machine learning models that can effectively detect fraudulent transactions with high accuracy while reducing false positives. This goal involves tackling the inherent class imbalance using diverse sampling methods, pinpointing key features signaling fraud, and setting up a strong foundation for fraud detection. The research utilizes a thorough approach involving data preprocessing, the application of class balancing methods like under sampling, oversampling, and SMOTE, and subsequently building and assessing machine learning models.

Furthermore, the research seeks to provide practical insights and recommendations for financial institutions to enhance their fraud detection capabilities. By combining advanced analytical techniques with domain knowledge, this study aims to contribute to the ongoing development of more effective fraud prevention systems in the financial sector.

2. Description of the Dataset and Fraud Data Analytics Method

2.1 Dataset Overview

This study utilizes a modified version of the IEEE-CIS Fraud Detection dataset (IEEE-CIS, 2019). The dataset consists of 100,000 financial transactions encompassing 101 features that mirror genuine e-commerce transaction behaviors. It demonstrates a notable class imbalance, with 88,682 valid transactions and 11,318 fraudulent transactions, mirroring the common distribution observed in real-world fraud instances. The analytical approach consists of three major phases: dataset preparation, handling class imbalance, and model development.

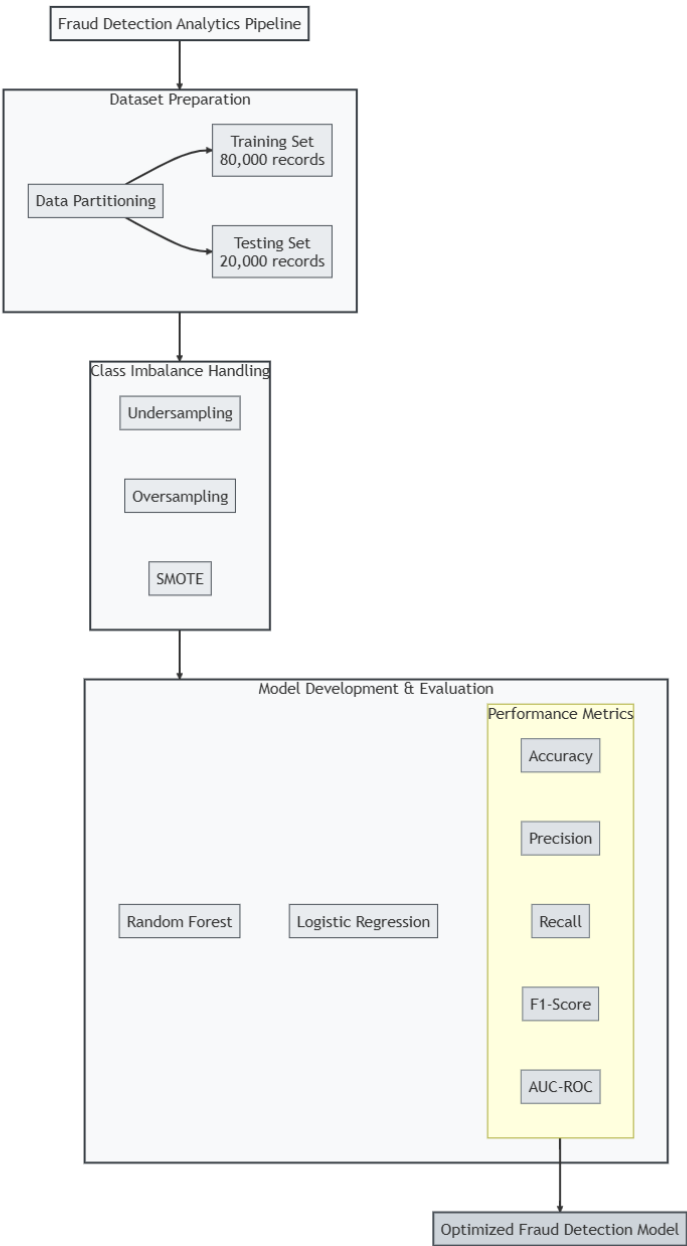


Fig.1 Data Analytics Pipeline

2.2 Dataset Preparation

The initial dataset was partitioned using an 80:20 split ratio, creating a training set of 80,000 transactions and a test set of 20,000 transactions. This split was implemented using random sampling with a fixed seed (123) to ensure reproducibility. The distribution of fraud cases was maintained proportionally across both sets, with the training set containing 70,896 legitimate and 9,104 fraudulent transactions, while the test set comprised 17,786 legitimate and 2,214 fraudulent transactions.

To mitigate the substantial class imbalance, three distinct techniques were implemented:

- *Under sampling*: The majority class (legitimate transactions) was reduced to match the minority class size, resulting in balanced but smaller dataset.
- *Oversampling*: The minority class (fraudulent transactions) was replicated to match the majority class size, maintaining all original data while achieving balance.
- *SMOTE (Synthetic Minority Over-sampling Technique)*: This advanced approach generated synthetic fraud cases based on the characteristics of existing fraudulent transactions. The process involved:
 - Converting categorical variables to numeric format
 - Handling missing values through median imputation
 - Creating synthetic samples using K=5 nearest neighbors
 - Generating a balanced dataset while preserving data integrity

2.3 Model Development

In this study, we employed two distinct machine learning algorithms, selected based on their complementary characteristics and demonstrated efficacy in financial fraud detection applications.

The primary methodology implemented was Random Forest, an ensemble learning algorithm that synthesizes multiple decision trees to construct a robust classification framework (Breiman, 2001). The selection of this methodology was predicated on several theoretical and empirical considerations. Random Forest demonstrates superior capability in capturing non-linear relationships inherent in transaction data, a critical factor given the intricate patterns characteristic of fraudulent activities. The algorithm's intrinsic feature importance quantification mechanism facilitates the identification of salient transaction attributes indicative of fraudulent behavior. Furthermore, Random Forest exhibits efficient handling of high-dimensional feature spaces, particularly pertinent to our dataset's multifaceted nature. The ensemble architecture inherently mitigates overfitting through bootstrap aggregation, thereby enhancing generalization performance on novel transaction instances.

As a comparative baseline, we implemented Logistic Regression, a well-established statistical learning method. Despite its relative algorithmic simplicity, Logistic Regression serves as a fundamental benchmark and offers distinct methodological advantages. The model's primary strength lies in its mathematical interpretability, a crucial consideration in financial domains where algorithmic decisions necessitate transparent explanation to regulatory entities. The probabilistic outputs generated by Logistic Regression enable flexible threshold adjustment for fraud classification. Notably, while structurally straightforward, the model demonstrates substantial computational efficiency in processing large-scale financial datasets and maintains widespread adoption in empirical studies due to its consistent performance characteristics.

This dual-methodological approach enables the exploitation of complementary algorithmic strengths: the sophisticated pattern recognition capabilities inherent in Random Forest, coupled with the interpretability and computational efficiency of Logistic Regression. The methodology synthesis establishes a robust analytical framework while maintaining the requisite transparency for financial applications. Both models were trained on the SMOTE-balanced dataset and evaluated using a comprehensive set of metrics including accuracy, precision, recall, F1-score, and AUC-ROC curves.

3. Results and Model Performance Analysis

The empirical results demonstrate the effectiveness of our dual-model approach in fraud detection, with both models showing distinct performance characteristics after training on the SMOTE-balanced dataset.

3.1 Class Imbalance Resolution

Our initial data analysis revealed a significant class imbalance in the dataset, necessitating the implementation of balanced learning techniques. We evaluated three distinct approaches to address this imbalance:

```
## Method 1: Undersampling
undersampled_majority <- majority_class[sample(nrow(majority_class), nrow(minority_class)), ]
undersampled_data <- rbind(undersampled_majority, minority_class)

## Method 2: Oversampling
oversampled_minority <- minority_class[sample(nrow(minority_class), nrow(majority_class), replace = TRUE), ]
oversampled_data <- rbind(majority_class, oversampled_minority)
```

After careful evaluation, we selected the SMOTE (Synthetic Minority Over-sampling Technique) algorithm as our primary approach. This decision was driven by several critical considerations. Unlike traditional undersampling, which potentially discards valuable majority class information, or simple oversampling, which may lead to overfitting due to exact

replication of minority instances, SMOTE generates synthetic examples in the feature space through interpolation (Chawla et al., 2002). This synthetic sample generation preserves the statistical properties of the minority class while introducing beneficial variations. Furthermore, SMOTE's approach of creating synthetic samples along the line segments joining nearest minority class neighbors helps prevent the overfitting problems associated with simple oversampling, while maintaining the full information content of the majority class.

```
# 1. Data Preparation for SMOTE
X <- as.data.frame(train_data[, -which(names(train_data) == "isFraud")]) # Features
Y <- train_data$isFraud # Target variable

# 2. SMOTE Application with Key Parameters
smote_result <- SMOTE(X, Y, K = 5, dup_size = 7)
smote_data <- smote_result$data
```

The SMOTE-based balanced dataset demonstrated effective resolution of the initial class imbalance problem. The original dataset exhibited a significant imbalance ratio, which was successfully addressed through SMOTE, resulting in balanced class distributions (70,896 non-fraudulent vs. 72,832 fraudulent cases). This balanced distribution contributed to the models' improved learning capabilities and reduced bias towards the majority class.

3.2 Model Training Evaluation

```
evaluate_model <- function(predictions, actual, probabilities = NULL) {
  conf_matrix <- confusionMatrix(predictions, actual)

  metrics <- c(
    Accuracy = conf_matrix$overall["Accuracy"],
    Precision = conf_matrix$byClass["Precision"],
    Recall = conf_matrix$byClass["Recall"],
    F1 = conf_matrix$byClass["F1"],
    AUC = if(!is.null(probabilities)) auc(roc(actual, probabilities)) else NA
  )

  return(metrics)
}
```

The model training phase was executed with meticulous attention to methodological rigor and comprehensive evaluation metrics. We implemented a sophisticated evaluation framework to ensure robust model assessment.

3.3 Random Forest Implementation and Feature Importance Analysis

The Random Forest model was implemented utilizing optimized hyperparameters (ntree = 500, mtry = sqrt(number of features)) based on established methodological principles (Breiman, 2001). The model demonstrated exceptional predictive performance, achieving an accuracy of 0.982 and an AUC score of 0.996 on the test dataset. The high precision (0.972) and recall (0.991) values indicate robust performance in both fraud detection and false positive minimization.

```
# Random Forest Implementation
rf_model <- randomForest(isFraud ~ .,
                          data = train_balanced,
                          ntree = 500,
                          mtry = sqrt(ncol(train_balanced)),
                          importance = TRUE)

# Model Evaluation
rf_pred <- predict(rf_model, test_balanced)
rf_evaluation <- evaluate_model(rf_pred, test_balanced$isFraud, rf_prob)
```

Feature importance analysis revealed that TransactionAmt (MeanDecreaseAccuracy = 71.91) and behavioral indicator C1 (MeanDecreaseGini = 6979.39) were the most influential predictors, suggesting the significance of both transaction characteristics and derived features in fraud detection.

3.4 Logistic Regression Analysis

The logistic regression model was implemented as a baseline comparison, employing a binomial distribution with a logit link function. The model achieved moderate performance metrics with an accuracy of 0.812 and an AUC of 0.884.

```
# Logistic Regression Implementation
lr_model <- glm(isFraud ~ .,
                data = train_balanced,
                family = "binomial")

# Model Evaluation
lr_prob <- predict(lr_model, test_balanced, type = "response")
lr_pred <- as.factor(ifelse(lr_prob > 0.5, "1", "0"))
lr_evaluation <- evaluate_model(lr_pred, test_balanced$isFraud, lr_prob)
```

3.5 Model Comparison and Recommendations

	Random Forest	Logistic Regression
Accuracy	0.9815279	0.8116955
Precision	0.9721020	0.8157388
Recall	0.9912416	0.8018498
F1	0.9815785	0.8087347
AUC	0.9959672	0.8840700

Table.1 Model Comparison

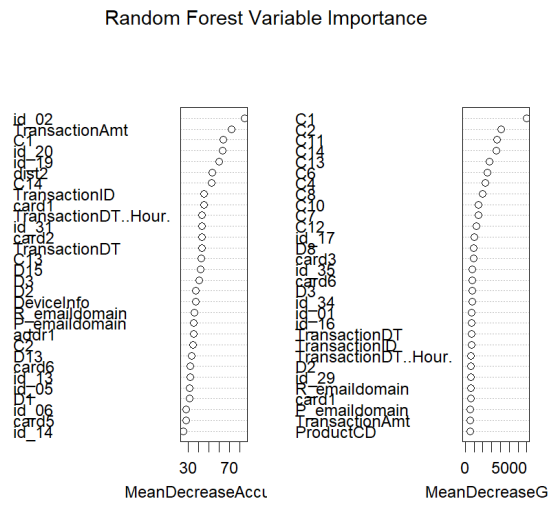


Figure.2 Random Forest Variable Importance

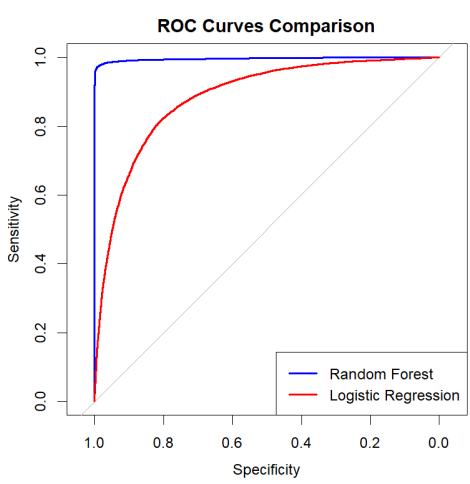


Figure.3 ROC Curves Comparison

A comprehensive evaluation of the implemented models reveals distinct performance characteristics (Table 1). The Random Forest model demonstrated superior predictive capabilities across all evaluation metrics, achieving an accuracy of 0.982 and an exceptional AUC score of 0.996. This performance significantly outperformed the logistic regression baseline, which achieved an accuracy of 0.812 and an AUC of 0.884.

The Random Forest model's high precision (0.972) coupled with outstanding recall (0.991) suggests robust fraud detection capabilities while maintaining a low false positive rate. This balance is crucial in practical applications where false positives can lead to customer dissatisfaction and operational inefficiencies.

3.6 Model Limitations

- Potential overfitting despite SMOTE balancing
- Computational intensity of Random Forest in real-time applications
- Limited interpretability compared to logistic regression

4. Recommendation

The recommendations encompass both technical implementation and essential non-technical controls for comprehensive fraud prevention. For technical implementation, the Random Forest model should serve as the primary detection mechanism, with Logistic Regression maintained as validation backup. Real-time monitoring systems should focus on key predictive features, with threshold-based alerts for suspicious patterns. Regular model retraining and sophisticated feature engineering should be implemented, combining machine learning capabilities with traditional rule-based systems for comprehensive coverage.

From a risk management and governance perspective, we recommend implementing a three-lines-of-defense model incorporating operational controls, risk management oversight, and internal audit. This should be supported by clear escalation procedures, regular risk assessments, enhanced transaction monitoring thresholds, and strengthened KYC procedures. Internal controls should include segregation of duties in transaction processing, mandatory dual control for high-risk transactions, regular compliance reviews, and comprehensive documentation of override decisions. capital development is crucial, requiring comprehensive staff training on fraud indicators, regular updates on emerging patterns, and establishment of cross-functional fraud response teams. Clear communication channels and performance metrics should be established to ensure effectiveness. External collaboration through information sharing with other financial institutions, engagement with regulatory bodies, and partnerships with cybersecurity experts will further enhance fraud prevention capabilities. This integrated approach ensures robust fraud detection while maintaining operational efficiency and regulatory compliance.

Summary

This comprehensive study developed an effective fraud detection framework combining advanced analytics with robust operational controls. The technical analysis of 100,000 financial transactions utilized machine learning techniques, addressing class imbalance through SMOTE sampling. The Random Forest model achieved 98.2% accuracy and 0.996 AUC score, outperforming Logistic Regression (81.2% accuracy, 0.884 AUC). Transaction amount and behavioral indicator C1 were identified as crucial predictive features. The study's success relies on integrating this technical solution with strong governance frameworks, internal controls, and human capital development. The recommended multi-layered approach combines automated detection with manual oversight, supported by clear policies, procedures, and regular training. This comprehensive strategy ensures effective fraud prevention while maintaining operational efficiency and regulatory compliance.

Reference

Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.

IEEE-CIS. (2019). IEEE-CIS Fraud Detection. Kaggle Competition Dataset. Retrieved from <https://www.kaggle.com/c/ieee-fraud-detection/data>

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. *Computers & security*, 57, 47-66.

Appendix 1. Random Forest Feature Importance

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini	D7	12.933513	22.447496	20.704800	123.252303
TransactionID	34.884854	37.365342	45.581115	689.247589	D8	18.749501	34.414845	21.816479	946.413243
TransactionDT	33.067035	43.889886	43.054855	692.765196	D9	12.030678	48.194209	16.733453	236.868566
TransactionDT..Hour	33.749792	40.817563	43.652833	666.865836	D10	8.785498	15.400613	13.996333	83.991289
TransactionAmt	54.006843	60.793650	71.907005	568.875911	D12	6.055969	18.938245	8.064197	273.710395
ProductCD	15.830910	17.403315	18.123600	556.483904	D13	17.759032	35.064410	33.234883	212.766047
addr1	17.906797	37.394764	35.234313	187.285161	D14	13.767576	25.584742	21.915753	185.540583
addr2	10.921628	24.948450	24.627491	43.438705	D15	31.656731	35.831517	41.737517	192.901449
dist2	27.680794	49.058108	53.047108	195.203748	M4	12.621115	12.460323	13.409861	544.932040
card1	28.250076	48.638807	45.199446	607.920483	V310	14.102345	19.278045	18.349778	183.412594
card2	28.280898	43.967164	43.305117	538.387150	V311	11.840169	18.229415	16.038058	33.360251
card3	13.646817	19.636159	17.176057	933.609279	V312	11.820550	18.875316	14.282190	111.911933
card4	14.117710	35.723888	21.174707	220.123817	V313	11.377004	25.381247	13.832707	102.227249
card5	18.058549	34.254728	27.882665	370.403336	V314	13.259849	22.388445	17.534847	117.107980
card6	25.333395	38.150431	32.305444	822.363190	id_01	13.734320	35.472859	22.548066	716.081223
P_emaildomain	26.689395	38.295263	35.489293	573.775230	id_02	65.088200	61.810435	84.723952	437.536904
R_emaildomain	26.274879	35.734926	36.147881	623.978650	id_03	14.018014	27.953429	18.279077	81.087447
C1	38.508529	64.533538	64.267956	6979.389052	id_04	13.919909	12.563040	17.479974	9.864744
C2	26.726094	31.520485	34.410567	4066.925969	id_05	20.874894	37.907323	31.642748	283.697556
C3	11.775886	7.824405	8.991947	7.495277	id_06	21.912547	28.718750	28.088603	300.125153
C4	16.129383	20.244606	20.323048	2267.169829	id_07	8.100192	13.466701	14.714356	25.138057
C5	0.000000	0.000000	0.000000	0.000000	id_08	8.202623	19.983522	16.366644	25.149979
C6	16.815297	19.684959	20.505470	2500.962295	id_09	15.611569	22.034961	19.136161	113.209088
C7	17.607203	16.654880	20.235220	1512.517247	id_10	9.196563	15.813186	15.911778	12.732216
C8	18.629355	19.865996	21.151964	1981.384911	id_11	12.741998	23.342279	17.944203	59.732050
C9	0.000000	0.000000	0.000000	0.000000	id_12	19.161955	16.638106	21.203441	473.849074
C10	14.824824	15.332445	16.279084	1526.837910	id_13	18.661457	39.544796	31.813942	217.405481
C11	20.168601	23.723988	24.627794	3624.378215	id_14	10.933596	28.946357	25.335122	127.031499
C12	16.257799	15.772639	18.684653	1257.282804	id_15	15.540459	22.504318	18.419502	423.858216
C13	37.963361	36.902379	42.651451	2748.959029	id_16	12.546688	16.786462	14.481404	699.733908
C14	47.151323	37.429288	52.808078	3538.353541	id_17	16.291963	14.762220	16.937885	1000.393899
D1	20.439557	33.419885	31.214183	501.590302	id_18	15.312227	21.928616	24.915054	99.999778
D2	23.938838	37.517491	37.605262	662.219647	id_19	27.270857	60.134240	60.329794	316.865589
D3	32.172324	39.209240	40.575206	767.253753	id_20	29.903313	67.192357	63.627515	350.620129
D4	10.811096	20.753878	15.552964	270.685984	id_21	7.135591	11.775428	13.632003	18.852418
D5	12.699694	18.649730	19.699992	93.661961	id_22	6.765472	10.023101	11.685438	5.168167
D6	10.610027	24.426549	15.245731	369.292877	id_23	12.232072	10.607533	14.588712	22.807675

id_24	6.756599 9.543708	11.034142	6.649893	id_33	15.059409 21.393496	17.815346	414.020459
id_25	6.776582 17.658186	16.182085	18.575117	id_34	17.387083 12.845364	20.140397	760.678377
id_26	10.895643 16.543014	17.240517	20.892893	id_35	14.794683 10.185767	15.460830	826.071587
id_27	9.106379 9.151726	12.814077	19.654212	id_36	9.732228 19.344671	21.206939	32.068579
id_28	19.809217 17.017693	22.052136	519.302796	id_37	13.691049 16.223785	19.366427	106.118380
id_29	16.040338 15.710109	17.996203	635.992003	id_38	12.025296 23.428746	19.518577	105.650870
id_30	17.009420 13.845847	20.741816	538.918381	DeviceType	16.148520 31.116230	20.640251	356.661231
id_31	21.457255 56.135072	43.423705	362.076781	DeviceInfo	26.465615 36.988317	37.415889	334.121449
id_32	16.300141 18.628934	18.820638	209.065222				

Appendix 2. Logistic Regression Coefficients

	Estimate	Std. Error	z value	Pr(> z)		
					D8	-2.347321e-03 7.487127e-05 -31.35143184 9.301491e-216
(Intercept)	-1.306447e+02	3.236720e+00	-40.36329253	0.000000e+00	D9	-5.253919e-02 3.350255e-02 -1.56821445 1.168311e-01
TransactionID	2.836794e-05	8.512847e-07	33.32367656	1.753318e-243	D10	-1.839330e-03 2.969900e-04 -6.19323983 5.893994e-10
TransactionDT	-1.082508e-05	8.118022e-06	-1.33346221	1.823802e-01	D12	4.145915e-03 4.777199e-04 8.67854892 4.008501e-18
TransactionDT..Hour.	3.550692e-02	2.922704e-02	1.21486515	2.244175e-01	D13	-2.817453e-03 3.192052e-04 -8.82646361 1.080375e-18
TransactionAmt	5.576493e-03	1.142908e-04	48.79213665	0.000000e+00	D14	3.495552e-04 8.885611e-05 3.93394695 8.356223e-05
ProductCD	-3.912044e-01	2.151514e-02	-18.18274316	7.070676e-74	D15	4.445064e-04 1.431139e-04 3.10596308 1.896604e-03
addr1	-2.740023e-04	1.275815e-04	-2.14766427	3.174044e-02	M4	-2.504575e-02 1.355576e-02 -1.84760952 6.465885e-02
addr2	3.191092e-02	2.685887e-03	11.88096311	1.486448e-32	V310	-4.173418e-03 3.152239e-04 -13.23953306 5.187031e-40
dist2	6.990064e-05	3.041916e-05	2.29791461	2.156665e-02	V311	-8.436029e-03 6.333301e-04 -13.32011433 1.768365e-40
card1	-3.182195e-05	1.747294e-06	-18.21213341	4.135148e-74	V312	1.033498e-03 3.798249e-04 2.72098399 6.508791e-03
card2	-1.363677e-03	5.277795e-05	-25.83800811	3.318613e-147	V313	3.740242e-04 4.105286e-04 0.91107946 3.622535e-01
card3	2.000482e-02	6.794886e-04	29.44098585	1.642163e-190	V314	3.173526e-03 3.688898e-04 8.60290856 7.772089e-18
card4	1.762985e-01	1.213856e-02	14.52383589	8.558387e-48	id_01	-2.349122e-02 6.194707e-04 -37.92143768 0.000000e+00
card5	-2.834586e-04	1.966990e-04	-1.44107822	1.495626e-01	id_02	-8.972668e-07 5.455072e-08 -16.44830542 8.624773e-61
card6	-7.496723e-01	1.762126e-02	-42.54362297	0.000000e+00	id_03	1.904963e-01 2.466557e-02 7.72316337 1.134776e-14
P_emaildomain	-1.071387e-01	5.918579e-03	-18.10210510	3.067181e-73	id_04	-3.895458e-01 2.301090e-02 -16.92875019 2.761821e-64
R_emaildomain	1.949259e-02	5.196267e-03	3.75126914	1.759417e-04	id_05	1.734605e-02 1.751551e-03 9.90325198 4.029547e-23
C1	-1.975651e-02	9.698473e-04	-20.37074607	3.040112e-92	id_06	1.013780e-03 5.758816e-04 1.76039634 7.834063e-02
C2	-1.433965e-02	8.111359e-04	-17.67847578	6.143470e-70	id_07	3.201835e-04 4.145233e-03 0.07724137 9.384315e-01
C3	-5.024076e+00	2.809783e-01	-17.88065521	1.668645e-71	id_08	-9.408072e-03 1.746491e-03 -5.38684428 7.170551e-08
C4	5.218057e-02	1.505627e-02	3.46570352	5.288461e-04	id_09	1.925514e-01 1.715879e-02 11.22173445 3.189527e-29
C6	-1.975210e-01	1.582605e-02	-12.48074682	9.509054e-36	id_10	4.555972e-02 9.760414e-03 4.66780639 3.044326e-06
C7	2.861594e-01	1.597861e-02	17.90890611	1.004911e-71	id_11	3.667169e-01 1.813557e-02 20.22086660 6.414641e-91
C8	5.933458e-02	2.549988e-03	23.26856900	9.228774e-120	id_12	1.404794e-01 3.140964e-02 4.47249146 7.731349e-06
C10	1.721245e-02	1.623649e-03	10.60109173	2.945218e-26	id_13	-6.929774e-04 7.672843e-04 -0.90315595 3.664431e-01
C11	4.686890e-01	1.399275e-02	33.49512266	5.675912e-246	id_14	2.222521e-03 1.347239e-04 16.49685708 3.864921e-61
C12	-3.766570e-01	1.436178e-02	-26.22635288	1.330496e-151	id_15	1.055892e+00 5.556544e-02 19.00266884 1.620889e-80
C13	-3.000674e-03	1.330115e-03	-2.25595107	2.407370e-02	id_16	7.167471e-01 3.074456e-02 23.31297637 3.274432e-120
C14	-4.780982e-01	7.924804e-03	-60.32933525	0.000000e+00	id_17	1.296939e-02 7.581048e-04 17.10764473 1.301571e-65
D1	-3.730999e-03	3.102907e-04	-12.02420632	2.651234e-33	id_18	3.689197e-02 1.007498e-02 3.66173936 2.505087e-04
D2	-2.779085e-03	3.560089e-04	-7.80622374	5.892693e-15	id_19	-1.979326e-04 6.723074e-05 -2.94407959 3.239167e-03
D3	2.080040e-03	3.633383e-04	5.72480223	1.035543e-08	id_20	2.494954e-04 6.159468e-05 4.05059932 5.108662e-05
D4	-2.724397e-03	4.793304e-04	-5.68375680	1.317676e-08	id_21	6.701548e-04 2.603560e-04 2.57399411 1.005320e-02
D5	-3.669376e-03	5.965485e-04	-6.15101070	7.699072e-10	id_22	4.443964e-03 6.978465e-03 0.63681103 5.242479e-01
D6	-2.586539e-04	2.516720e-04	-1.02774222	3.040711e-01	id_23	-2.894701e-01 7.142369e-02 -4.05285846 5.059562e-05
D7	-2.347655e-04	5.911210e-04	-0.39715299	6.912546e-01	id_24	-3.164662e-02 1.877984e-02 -1.68513799 9.196195e-02

id_25	9.338533e-04	5.152307e-04	1.81249561	6.990965e-02	id_33	1.454825e-04	2.685028e-04	0.54182839	5.879367e-01
id_26	1.044970e-02	1.511589e-03	6.91305622	4.743219e-12	id_34	1.772356e-01	1.751167e-02	10.12100174	4.458290e-24
id_27	1.783028e-01	2.061240e-01	0.86502697	3.870240e-01	id_35	-1.290979e+00	7.525869e-02	-17.15388415	5.878840e-66
id_28	4.946839e-01	2.300112e-01	2.15069492	3.150029e-02	id_36	-3.717308e-01	4.929295e-02	-7.54125682	4.654634e-14
id_29	-3.289250e+00	2.247497e-01	-14.63517056	1.675550e-48	id_37	-3.334829e-02	3.795186e-02	-0.87869981	3.795641e-01
id_30	1.101748e-01	9.144073e-03	12.04877039	1.968647e-33	id_38	2.889336e-01	2.449071e-02	11.79768221	4.012104e-32
id_31	-4.756871e-03	4.028197e-04	-11.80893389	3.509834e-32	DeviceType	4.042047e-01	2.447240e-02	16.51675677	2.779481e-61
id_32	1.250337e-01	5.822639e-03	21.47372030	2.741676e-102	DeviceInfo	-1.846340e-04	1.834772e-05	-10.06304890	8.046697e-24