

Linear Models for Data Science (STAT 5420) Project:

**Credit Card Fraud Detection Analysis**

Jiayi Zhang (T00752757)

Arpitha Thippeswamy (T00749833)

Shashank Manjunatha (T00728166)

Sree Aryan Sathyamurthy Parimala Priyadarshini (T00751318)

March 29th, 2025

**Table of Contents:**

|  |  |
| --- | --- |
| **Title** | **Page No.** |
| **Abstract** | **3** |
| 1. **Introduction** | **4** |
| 1. **Literature Survey: Related Work** | **5** |
| 1. **Data Preparation and Exploration** | **6** |
| * 1. **Dataset overview** |  |
| * 1. **Data Pre Processing** |  |
| 1. **Methodology** | **7** |
| * 1. **Feature Selection Using LASSO Regression** |  |
| **4.2 Handling Class Imbalance Using SMOTE** |  |
| **4.3 Logistic Regression as a GLM** |  |
| 1. **Result** | **9** |
| 1. **Conclusion** | **12** |
| 1. **References** | **13** |

**Abstract**

Credit card fraud detection is a critical challenge in financial security, requiring advanced statistical techniques to identify fraudulent transactions while minimizing false positives. This project analyzes a real-world credit card fraud dataset consisting of 284,807 transactions, of which only 0.172% (492 cases) are fraudulent. Due to this severe class imbalance, traditional classification methods may fail to detect fraudulent transactions effectively. To address this, we employ a combination of feature selection, statistical inference, and machine learning techniques to enhance fraud detection accuracy.

First, we implement Least Absolute Shrinkage and Selection Operator (LASSO) regression to perform feature selection, identifying the most significant predictors from the 28 anonymized PCA-transformed features. By applying L1 regularization, LASSO eliminates irrelevant variables, reducing model complexity and improving interpretability.

Next, we conduct statistical inference on model coefficients to assess the impact of selected features on fraud probability. This includes bootstrapped confidence intervals, hypothesis testing using Wald statistics, and odds ratio analysis, providing deeper insights into how specific transaction characteristics influence fraud detection.

Finally, we develop a Generalized Linear Model (GLM) using logistic regression, optimized for fraud classification. Given the dataset’s class imbalance, we apply stratified sampling and class weighting techniques to ensure the model remains sensitive to fraudulent cases. Model performance is evaluated using precision, recall, F1-score, and AUC-ROC metrics to ensure both effectiveness and reliability.

This study demonstrates how integrating statistical techniques with machine learning can significantly enhance fraud detection. By selecting the most relevant features, interpreting their impact, and developing a robust classification model, we provide a data-driven approach to financial fraud prevention. The findings contribute to the broader field of fraud detection, offering practical insights for real-world financial security applications.

**1. Introduction**

With the rapid growth of digital transactions, credit card fraud has become a major financial threat, resulting in billions of dollars in losses annually. Fraudulent transactions not only impact financial institutions but also cause inconvenience and financial distress to consumers. As cybercriminals continuously adapt their techniques, traditional rule-based fraud detection methods struggle to keep pace. These conventional methods often rely on predefined rules, such as transaction limits or geographical constraints, which can fail to detect sophisticated fraud patterns or lead to high false positive rates.[1]

To address these challenges, machine learning and statistical modeling have emerged as more effective approaches for fraud detection. These models analyze large-scale transaction data, recognize patterns, and differentiate between legitimate and fraudulent activity with higher accuracy. However, fraud detection poses unique challenges, such as extreme class imbalance (fraudulent transactions make up only a tiny fraction of all transactions), feature interpretability, and the need for real-time predictions. This study aims to leverage statistical methods to improve fraud detection accuracy while addressing these challenges.

This analysis aims to identify key transaction features that influence fraud detection and improve classification accuracy using statistical modelling techniques. Specifically, we will explore:

* Which variables contribute most significantly to fraud detection?
* How do selected variables influence fraud likelihood?
* How well does logistic regression perform in classifying fraudulent transactions?

To answer these questions, we employ the following statistical methods:

1. LASSO Regression for Variable Selection
2. Apply SMOTE transformation
3. Generalized Linear Model (GLM) – Logistic Regression for Classification

**2. Literature Survey: Related Work**

1. **A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, "Adaptive Machine Learning for Credit Card Fraud Detection: The Importance of Dynamic Sampling Strategies," in *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics*, Paris, France, 2015, pp. 1-8.**

This foundational work addresses the critical challenge of class imbalance in fraud detection datasets through innovative adaptive resampling techniques. The authors conduct extensive experiments comparing static and dynamic sampling approaches, demonstrating that adaptive methods significantly improve detection rates for rare fraud cases. Their proposed framework incorporates feedback mechanisms that continuously adjust the sampling distribution based on model performance metrics. The paper provides detailed analysis of computational trade-offs between different sampling frequencies and batch sizes.

1. **S. Ghosh and D. L. Reilly, "Neural Network Based System for Credit Card Fraud Detection: An Early Application of Artificial Intelligence in Financial Security," in *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, USA, 1994, pp. 621-626.**

As one of the earliest academic works on automated fraud detection, this pioneering paper demonstrates the application of neural networks to financial anomaly detection. The authors present a multi-layer perceptron architecture trained on historical transaction data to identify fraudulent patterns. The paper introduces fundamental concepts of behavioural profiling that remain relevant in modern fraud detection systems. Practical implementation challenges including feature engineering and model interpretability are thoroughly discussed.

1. **Y. Sahin, S. Bulkan, and E. Duman, "Cost-Sensitive Decision Tree Approach for Financial Fraud Detection: Optimizing the Trade-off Between Fraud Prevention and Operational Costs," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.**

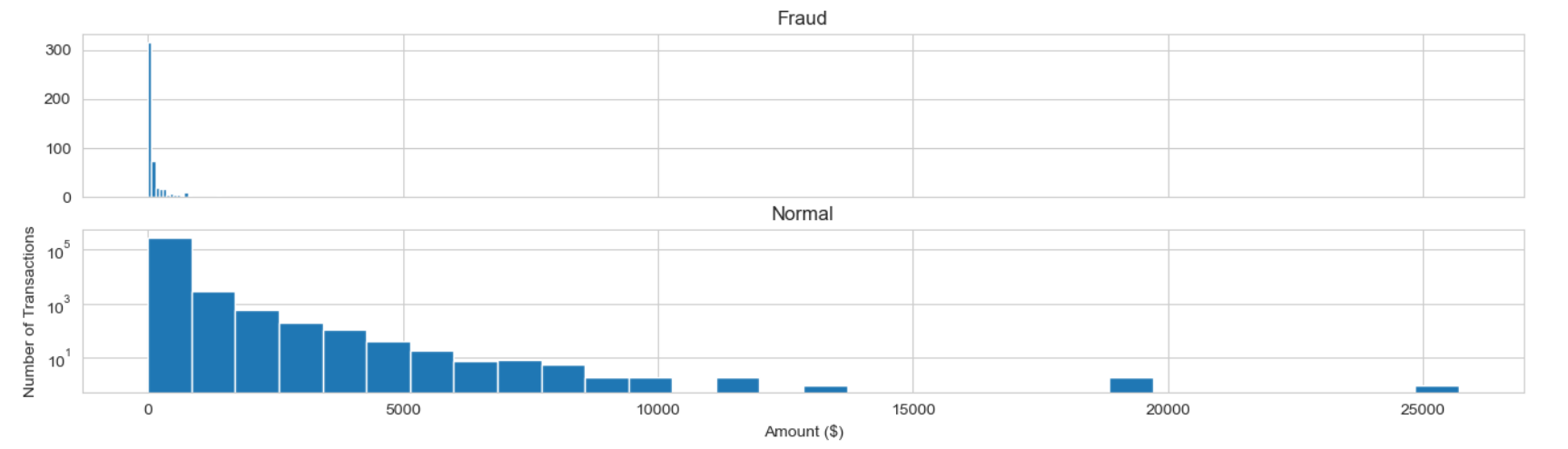
This paper presents a cost-sensitive decision tree model tailored for financial fraud detection. Unlike traditional models, it incorporates a cost matrix to account for the financial impact of misclassifications. The modified C4.5 algorithm helps minimize both fraud losses and operational expenses. Tested on real credit card data, the model proves more effective in balancing detection performance and cost-efficiency. The approach offers practical benefits for financial institutions aiming to optimize fraud prevention strategies.

**3. Data Preparation & Exploration**

**3.1 Dataset Overview**

The dataset contains 284,807 credit card transactions, each described by 31 numerical features. These include:

* **Time:** The number of seconds elapsed since the first transaction.
* **Amount:** The monetary value of the transaction.
* **V1–V28:** A set of anonymized features generated using Principal Component Analysis (PCA) to protect sensitive information.
* **Class (Target Variable):** A binary indicator where 0 represents a legitimate transaction and 1 indicates fraud.



**Fig.1.** Distribution on Fraud and Normal

With only 492 fraudulent transactions (0.172%) and mostly in low amount. The dataset exhibits extreme class imbalance, requiring specialized techniques to improve fraud detection accuracy.[2]

**3.2 Data Preprocessing**

To enhance model performance, several preprocessing steps are applied:

1. **Feature Engineering:** Since the Time feature is recorded in seconds, we derive an Hour variable to capture temporal transaction patterns, potentially revealing fraudulent behaviour trends.
2. **Normalization**: The Amount feature is standardized using z-score normalization to ensure all features contribute equally in statistical models, preventing dominance by variables with larger scales.
3. **Class Imbalance Handling:** Given the rarity of fraudulent cases, we use stratified sampling to maintain class proportions in training and testing sets, ensuring the model learns effectively from both legitimate and fraudulent transactions.

These preprocessing steps help prepare the dataset for accurate and interpretable fraud detection modelling.

**4. Methodology**

**4.1 Feature Selection Using LASSO Regression (Feature Engineering)**

**Idea**: Least Absolute Shrinkage and Selection Operator(LASSO) regression can introduce L1 regularization, penalizing large coefficients and shrinking some to exactly zero, effectively performing feature selection. These features of LASSO perfectly fit our need to eliminate the features. Since then, we opted for the L1 penalty instead of L2 regularization (Ridge regression) or a combination of L1 and L2 (Elastic Net regression).

LASSO regression solves the following optimization problem:

where:

A cross-validation approach is used to determine the optimal λ, ensuring the best trade-off between model complexity and prediction accuracy. LASSO reduces feature dimensionality, retaining only the most significant predictors, which are then used in subsequent analyses.

**Application:** In this essay, we use LASSO toexclude redundant or irrelevant predictors in 28 PCA-transformed features (V1–V28) and contain useful predictors for regression.

**4.2 Handling Class Imbalance Using SMOTE (Class Imbalance Handling)**

**Idea:** Since the dataset contains significantly fewer fraudulent transactions compared to legitimate ones, directly training a model would lead to biased results where the classifier predominantly predicts non-fraudulent transactions. To address this strong imbalance, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to our work.

**4.2.1 SMOTE Algorithm:**

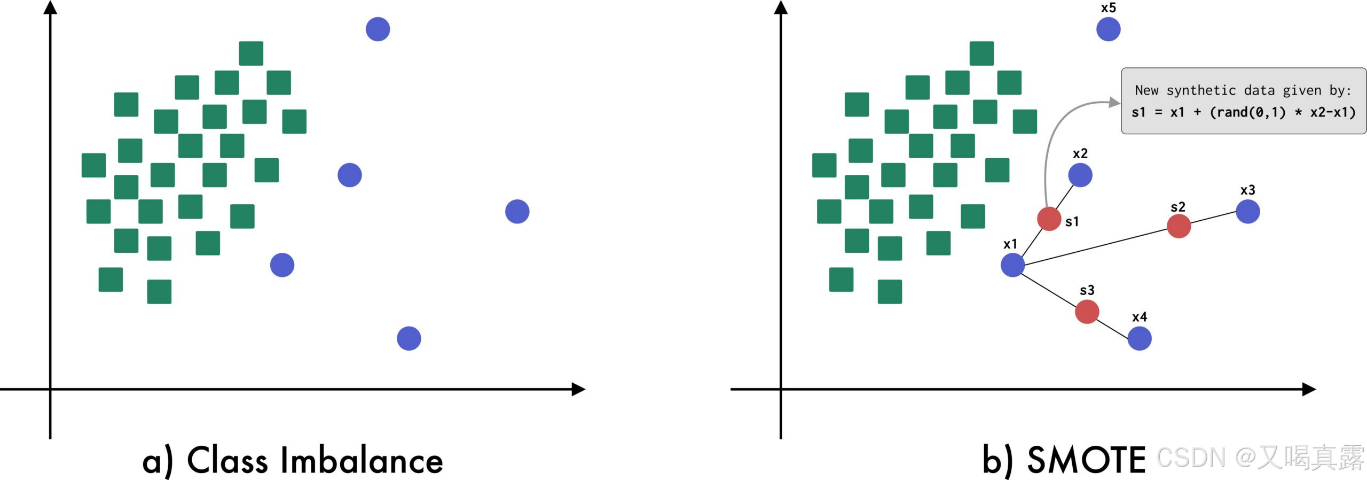
SMOTE is an oversampling method that generates synthetic data points for the minority class (fraudulent transactions) rather than duplicating existing ones. It creates new samples by interpolating between real fraudulent transactions. The algorithm has four steps:

1. Decide on which minority class should be picked

2. For each minority class instance (fraudulent transaction in this essay), its k-nearest neighbors are identified.

3. A random neighbor is selected, and a new synthetic instance is created along the line segment between the original and the selected neighbor.

4. The process is repeated until the minority class reaches the desired proportion in the dataset. (In this essay we set half-half is the desired proportion)



**Fig.2.** SMOTE Visual Description

Take an example, there are two categories of dots. Realized that green dots amount is far more than blue dots.

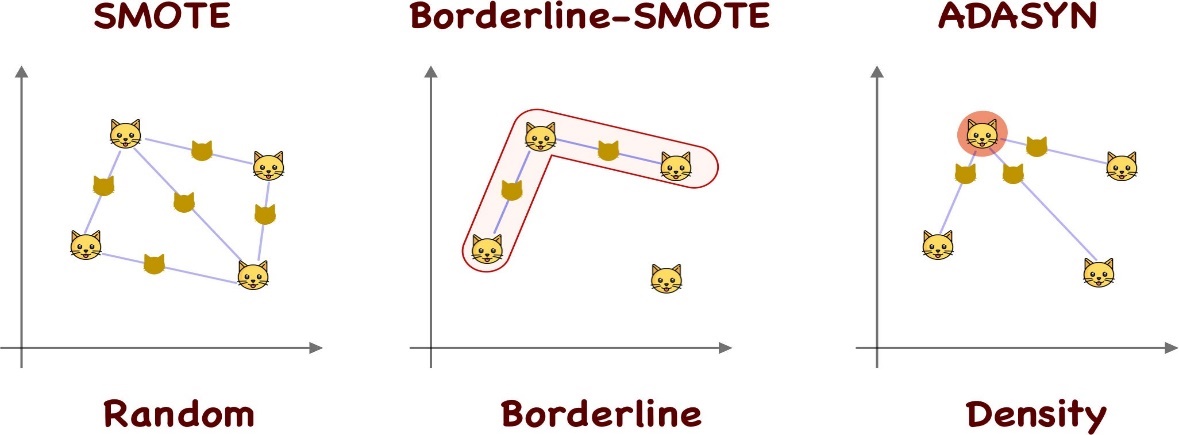
**Step 1**: So, use SMOTE on blue dots because it is minority.

**Step 2**: Then considered , identify that  are all its near neighbours.

**Step 3**: In this scenario, code randomly pick ,the new synthetic data sample is derived from the and the distance between and , which is |,multiple a random number between 0 and 1to create randomness.

**Step 4**: Iterate Step1 to Step 3, derive , until it reaches the satisfied need for research.

It is mentioned that **SMOTE** only works with continuous data (that is, it is not designed to generate categorical synthetic data), on the other hand, the synthetic data generated is *linearly dependent*, which can cause a bias in the data generated and consequently produce an overfitted model. For this reason, alternatives based on **SMOTE** have been proposed that aim to improve the limitations of the original **SMOTE** technique such as **Borderline-SMOTE** or **ADASYN.** [4]



**Fig.3.** Alternatives based on **SMOTE** visualization

**Application:** We will apply SMOTE to let fraud and normal data amount approximately half-half distribution using function package in Jupiter Notebook.

**4.3 Logistic Regression as a Generalized Linear Model (GLM) to fit the prediction model**

**Idea:** Since our target variable (fraud or non-fraud) is binary and it is hard to fit in the simple linear regression, when choosing GLM, logistic regression is perfect for classification of two categories and the algorithm give the clear boundary for more precise prediction. Unlike linear regression, logistic regression models the probability of fraud using the logit function, ensuring predicted probabilities fall within (0,1). [3]

The model is given by:

**Application:** We will apply logit transformation ensures a nonlinear relationship between predictors and fraud probability, and then make regression for prediction in nine different thresholds.

**5. Result:**

**Step 1: LASSO Selection**

The primary objective of this study is to develop a logistic regression model that classifies transactions as either normal or fraudulent, using "Class" as the target variable. However, a key challenge lies in the large number of predictor variables (28 in total). Given that LASSO regression (L1 regularization) can effectively eliminate less informative variables by shrinking their coefficients to exactly zero, we use the

By applying LASSO, we identified and removed 10 variables with minimal contribution to the model's predictive power (V1, V2, V3, V7, V11, V12, V18, V19, V24, V26) retaining rest of18 key features for further analysis.

The resulting logistic regression model achieved an accuracy of **0.999**, which is highly satisfactory. Moreover, both precision and recall scores are relatively high across categories (**1.00, 1.00** for normal transactions and **0.83, 0.63** for fraudulent transactions). These results provide confidence in the model’s effectiveness in distinguishing between normal and fraudulent transactions.

**Step 2: SMOTE Transformation**

Firstly, we use the SMOTE method which is mentioned in the essay to balance the amount of both class of data.

The method turns 19 dimension of the features (including 18 key features and amount features) into a 50-50 distribution for positive(normal) and negative(fraud). Thus, we can treat the original problem as a binary classification prediction problem and make step 3 Logistic regression less bias.

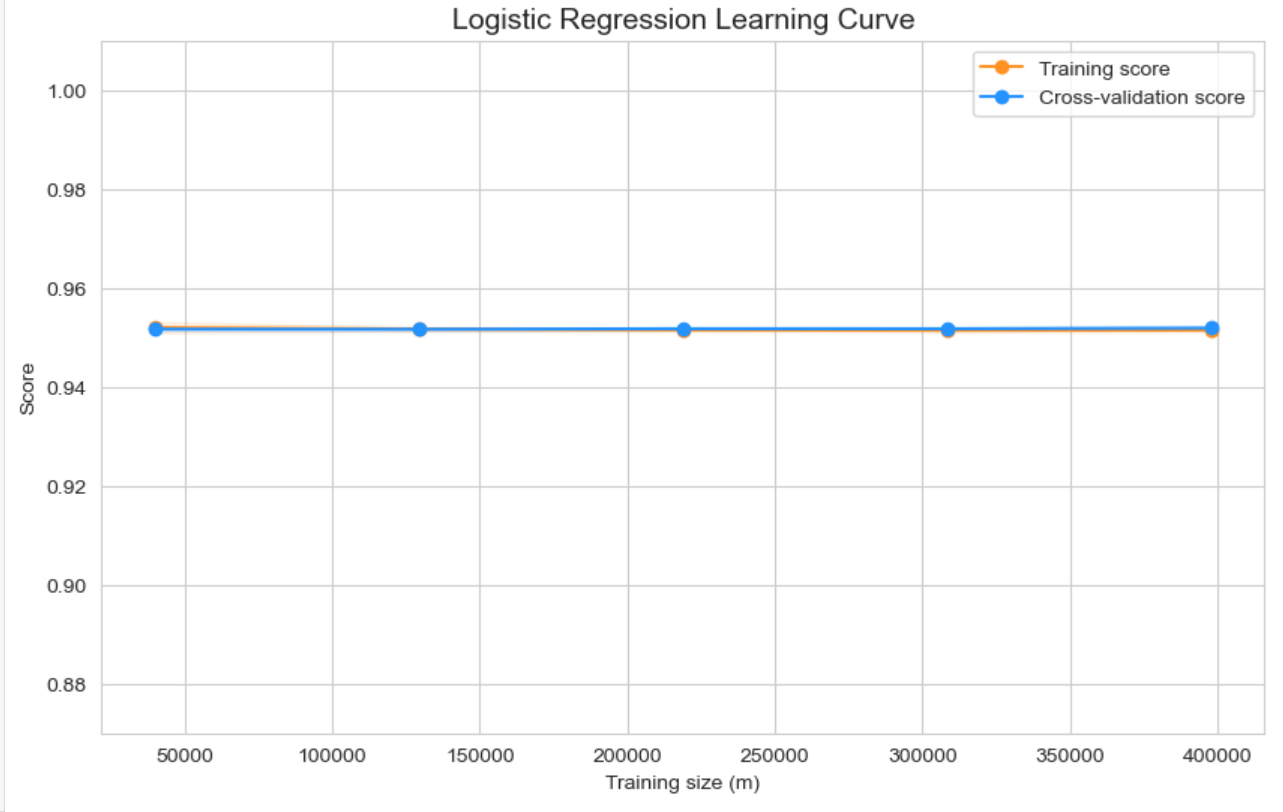
**Step 3: Logistic Regression:**

In this step, we mainly evaluate two features: precision and Recall score.

Precision: The proportion whether the model predict fraud as right category correctly. Reflect the precise degree of the model.

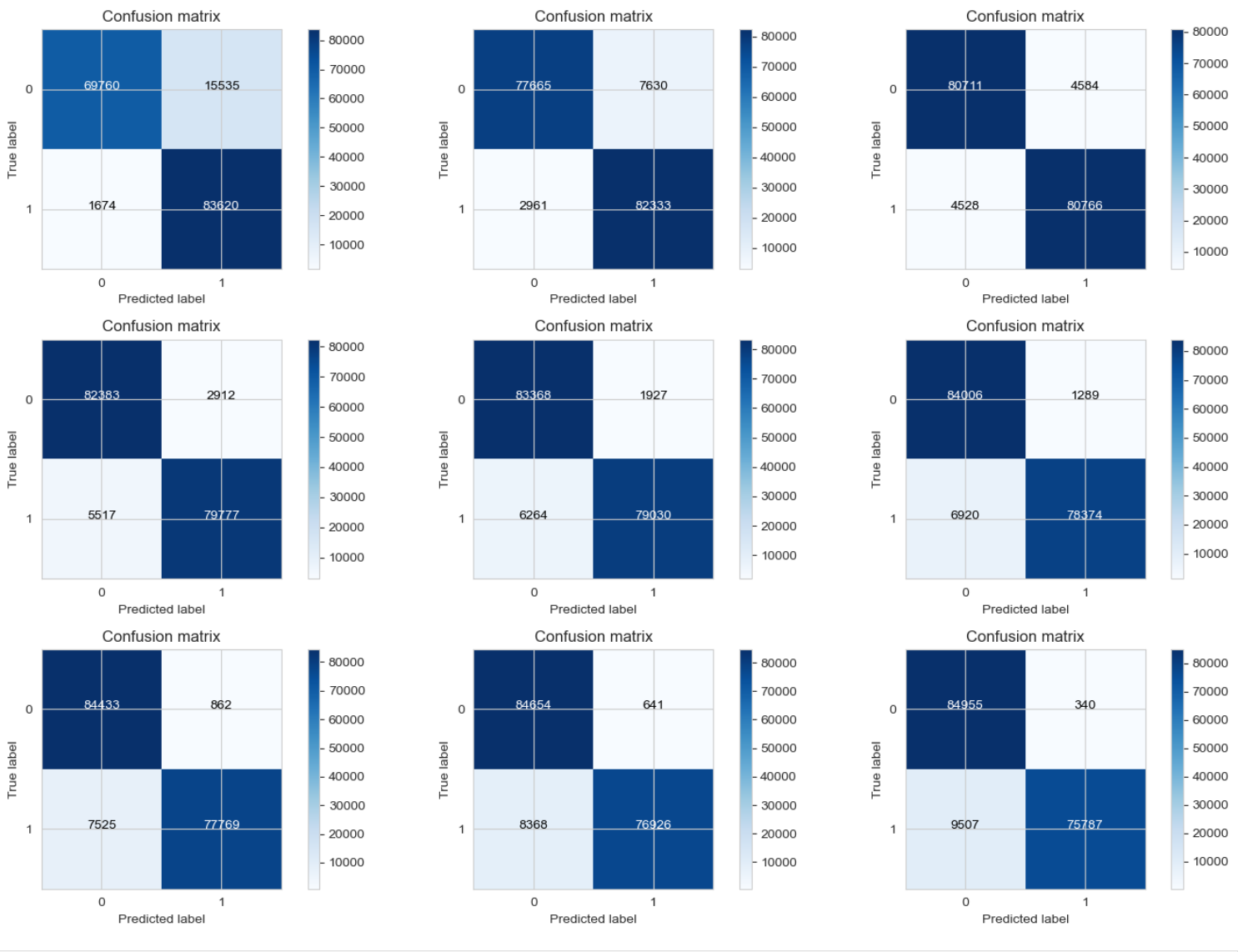
Recall: The proportion whether the model give the actual fraud cases correctly identified, reflect the accuracy on grasp any possible fraud. [1]

Then, after training the logistic model and making prediction. The resulting logistic regression model achieved an accuracy of **0.95**, which is highly satisfactory. Moreover, both precision and recall scores are relatively high across categories (**0.98,0.93** for normal transactions and **0.93,0.98** for fraudulent transactions). These results show the well adapt on the test data for the model.



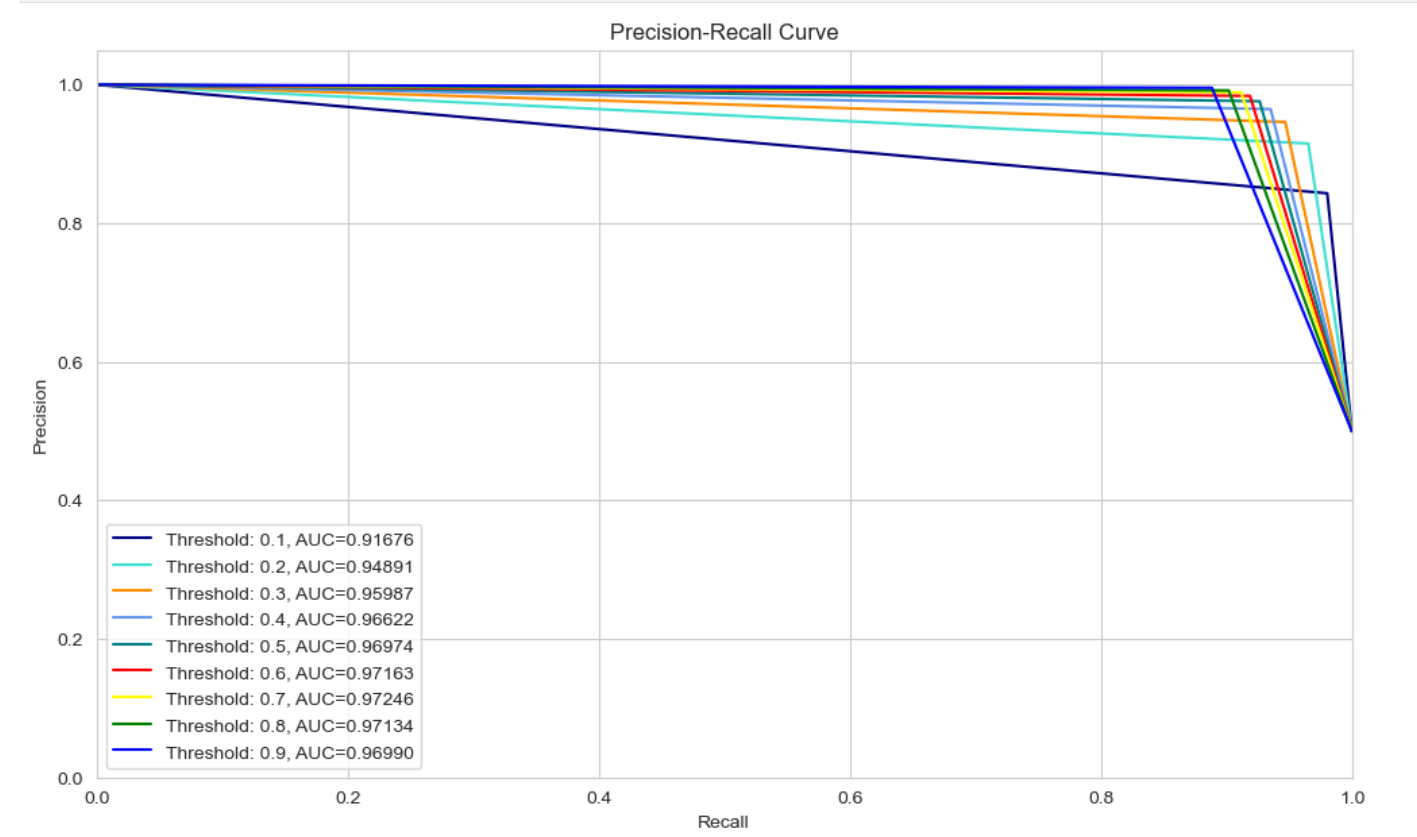
**Fig.4.** Logistic Regression Learning Curve

The logistic regression learning curve for training and the cross validation is fitted closely at approximate 0.95 high level score, showing the magnificent model performance.



**Fig.5.** Confusion Matrix on different threshold

From the summary of the confusion matrix, we can easily observe that for threshold values ranging from 0.1 to 0.9, the recall score remains high, indicating that the model rarely makes Type I and Type II errors in its predictions. As the threshold increases, the recall rate gradually declines, which is a controversial trend



**Fig.6.** Precision-Recall Curve

From the Precision-Recall Curve, we not only gain a visual representation of the relationship between precision and recall but can also draw the following conclusions:

* Precision and recall are two conflicting variables. There is a balance depend on the scenario.
* As observed from the confusion matrix and the Precision-Recall Curve (PRC), a lower threshold results in a higher recall value, meaning the model can identify more instances of credit card fraud. However, this comes at the cost of an increased number of false positives.
* Conversely, as the threshold increases, the recall value gradually decreases while the precision value improves, thereby reducing the number of false positives.
* By adjusting the model’s threshold, the intensity of fraud detection can be controlled. A lower threshold should be set if the goal is to identify more fraudulent credit card transactions, whereas a higher threshold should be chosen to minimize false positives.

Finishing step1 to step 3, the general model for Credit Card Fraud Detection is built up. Step 1 and Step 2 basically give a prepare for regression and Step 3 give the result.

**6. Conclusion:**

In this study, we selected Generalized Linear Models (GLM) as our baseline model due to its greater interpretability compared to black-box models such as ensemble learning or random forests. The transparent nature of GLM facilitates a more explicable parameter selection process, enhancing model comprehension and decision-making.

Furthermore, we prioritized the recall rate over the precision rate in our analysis. Given the objective of identifying fraudulent transactions to mitigate potential financial losses, maximizing recall ensures that a greater number of fraudulent activities are detected. While this decision inevitably leads to a significant decline in precision, the trade-off is justified by the need for a more comprehensive fraud detection system.

Lastly, our findings underscore the critical role of feature engineering prior to model prediction. Without an effective feature engineering process, handling the 28-dimensional feature space results in excessive computational resource consumption. Properly engineered features improve model efficiency and performance, reinforcing the necessity of this step in predictive modelling.

**7. References:**

**[1]** A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, "Adaptive Machine Learning for Credit Card Fraud Detection: The Importance of Dynamic Sampling Strategies," in *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics*, 2015, pp. 1-8.

**[2]** S. Ghosh and D. L. Reilly, "Neural Network Based System for Credit Card Fraud Detection: An Early Application of Artificial Intelligence in Financial Security," in *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, 1994, pp. 621-626.

**[3]** Y. Sahin, S. Bulkan, and E. Duman, "Cost-Sensitive Decision Tree Approach for Financial Fraud Detection: Optimizing the Trade-off Between Fraud Prevention and Operational Costs," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916-5923, 2013.

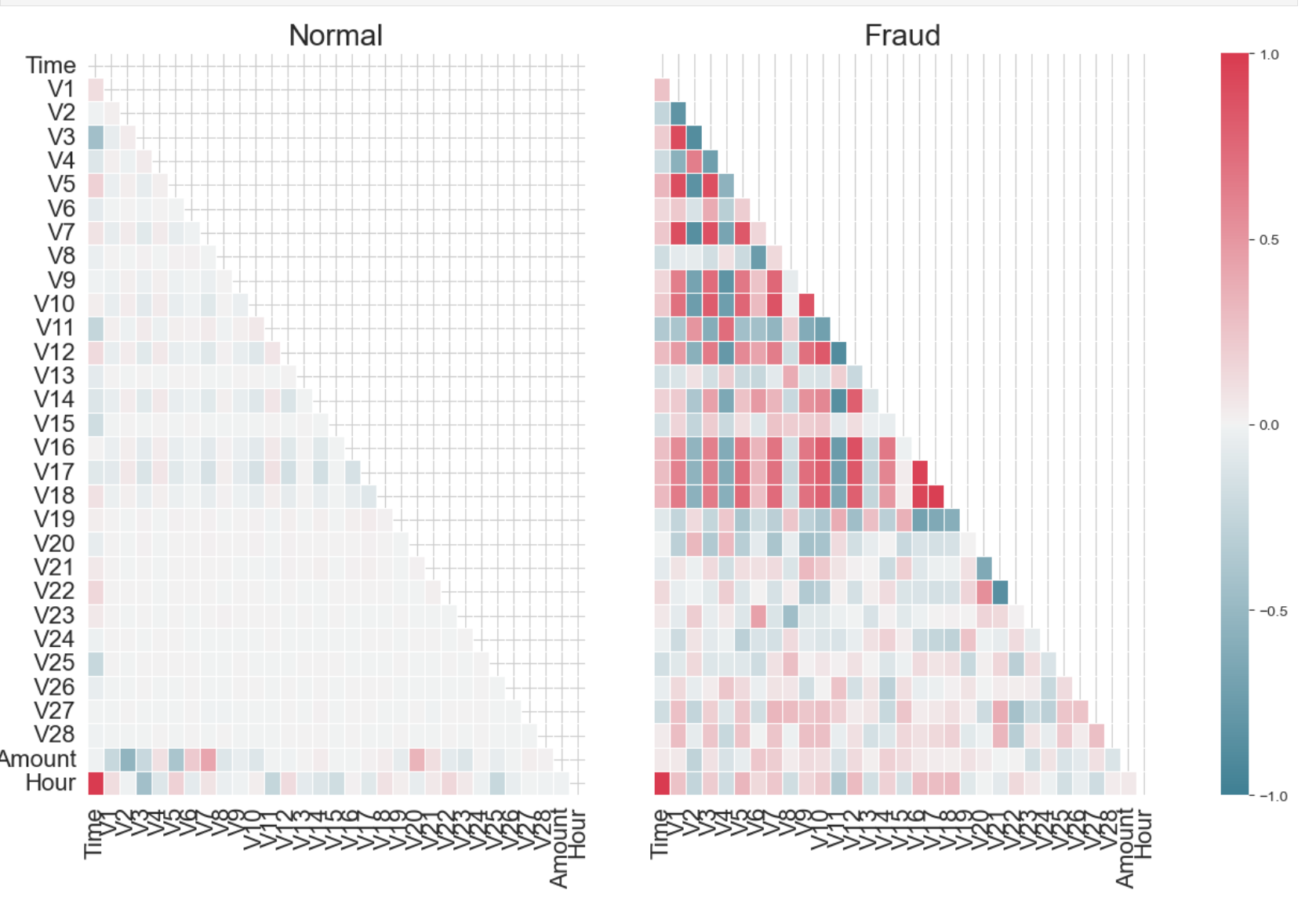
**[4]** F. López, "SMOTE: Synthetic Data Augmentation for Tabular Data. An exploration of SMOTE and some variants like Borderline-SMOTE and ADASYN," Mar. 1, 2020.

**[5]** Citation: *[1] Google, "Google Search," [Online]. Available:* [*https://www.google.com*](https://www.google.com)*. [Accessed: March 2025].*

**[6]**Citation: *OpenAI, "ChatGPT, an AI language model," OpenAI, [Online]. Available:* [*https://chat.openai.com/*](https://chat.openai.com/)*. [Accessed: March 2025].*

**Appendix:** Code: <https://github.com/zhajiyoo/Linear-Model-Project.git>

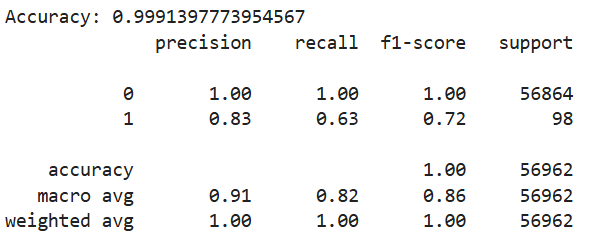
**Appendix: Useful Output**



**Fig.1.** The heatmap between features



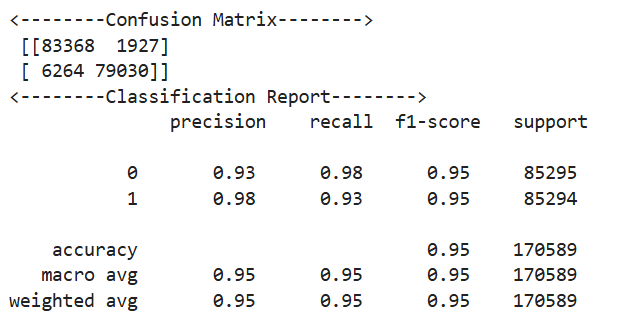
**Fig.2.** Chosen traits and deleted traits



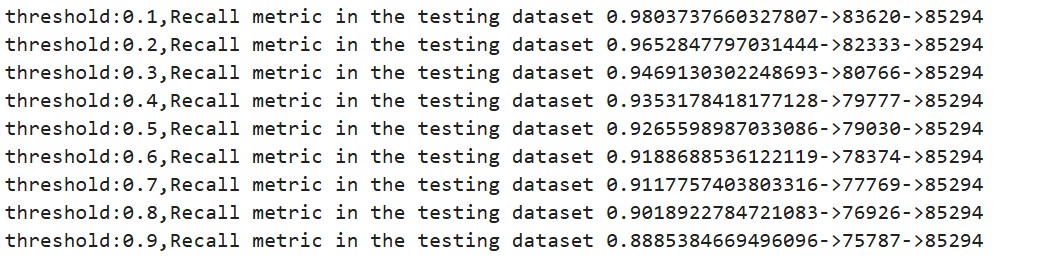
**Fig.3.**Summary on LASSO



**Fig.4.** Summary on SMOTE



**Fig.5.** Summary on Logistic regression



**Fig.6.** Recall rate on different threshold