Optimizing Sample Size for Logistic Regression Analysis of Credit Card Fraud Data

Austin Phillips & Ivo Schossow

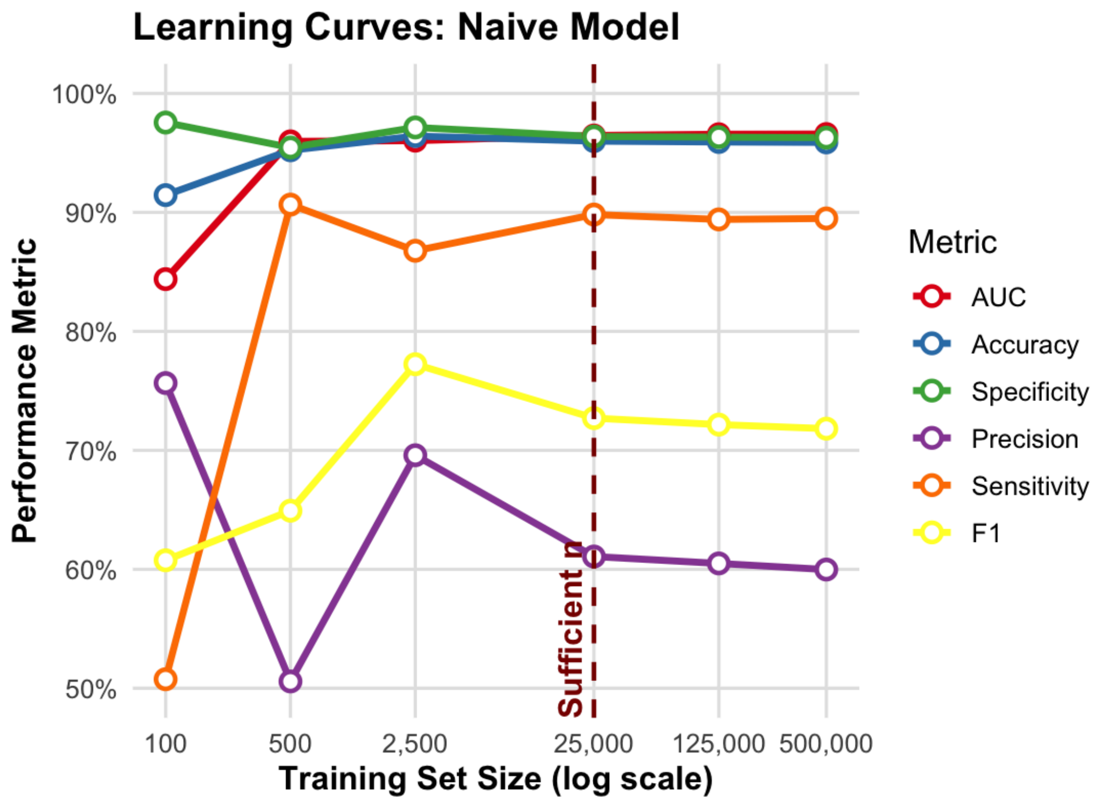
The world of ecommerce has helped push humanity into an age of unfettered success, as the ability to transact goods and services instantaneously has transformed our global economy. However, this advancement has also brought upon us unforeseen risks, one of which being the advent and popularization of credit card fraud. Detecting and halting these fraudulent claims has been at the forefront of the banking industry, as developing better models can help prevent these theft attempts (Kou et al., 2004; Bhattacharyya et al., 2011; Sahin & Duman, 2011). Creating these models in a time-efficient manner is imperative, as models need to optimize for both accuracy and speed. Therefore, we aim to solve the question: What is the ideal size of a fraud detection training model that balances both accuracy and computational time?

To test our model, we would need to find a dataset. What seems like a somewhat trivial task becomes difficult, as the public is not privy to real data, as it contains confidential information. Our solution for this issue was to use simulated data, which is representative of what information banks have when trying to identify fraud. This dataset includes 1 million total observations, and contains the following continuous variables: distance from home, distance from last transaction, ratio to median purchase price (Narayanan, 2022). And the following binary variable: repeat retailer, used chip, used pin number, online order, and finally whether the data was truly fraud.

The first thing that was required was to divide the data into a training set and a testing set. The first 500,000 were the training set, while the other half was the testing set. Since our goal is to identify what sample size is sufficient, we then will have to subdivide our training set into different sections, and compare the test results of each sample size. We broke our training data into 6 sizes: 100, 500, 2500, 25000, 125000, and 500000. We increased our sample size approximately multiplicatively (between 4x & 10x), as it is well understood that in most types of modelling, increasing sample sizes has diminishing returns on model effectiveness (John & Langley, 1996).

We will be using 3 different types of modeling to analyze the data. First, we will create a naive model, where all of our variables are utilized in creating a logistic regression model (GLM), which is a well-established and widely used statistical method for credit card fraud classification (Sahin & Duman, 2011; Bhattacharyya et al., 2011). Next, we will test to make sure our model assumptions are correct (multicollinearity, log-odds relationship, etc), and then make the necessary transformations to correct the model. Finally, we will introduce a Generalized Additive Model (GAM). With these three models finished, we will test all 3 against our testing dataset, and determine performance based on run time and 6 test statistics: AUC (Area Under the Curve), Accuracy, Sensitivity, Precision, Specificity, and F-Score (F-Score refers to a harmonic mean of Specificity and Precision) (Bhattacharyya et al., 2011; Teh et al., 2018).

To create our baseline model, we went ahead and created our naive model. (see Figure 1)
Figure 1:



Note: The graph above shows the test statistics and how they change as a measure of sample size. Test statistics level out after an n of 25,000. The final test statistics using the full training set were as follows: AUC - 96.6%, Accuracy - 95.9%, Sensitivity - 89.8%, Precision - 60.0%, Specificity - 96.3%, F1 - 71.8%.

The data above gives us our baseline model. Using this model, we see that our test statistics level out after a sample size of 25,000. Using this as a baseline, we can see if our other models also have a similar “sufficient n”.

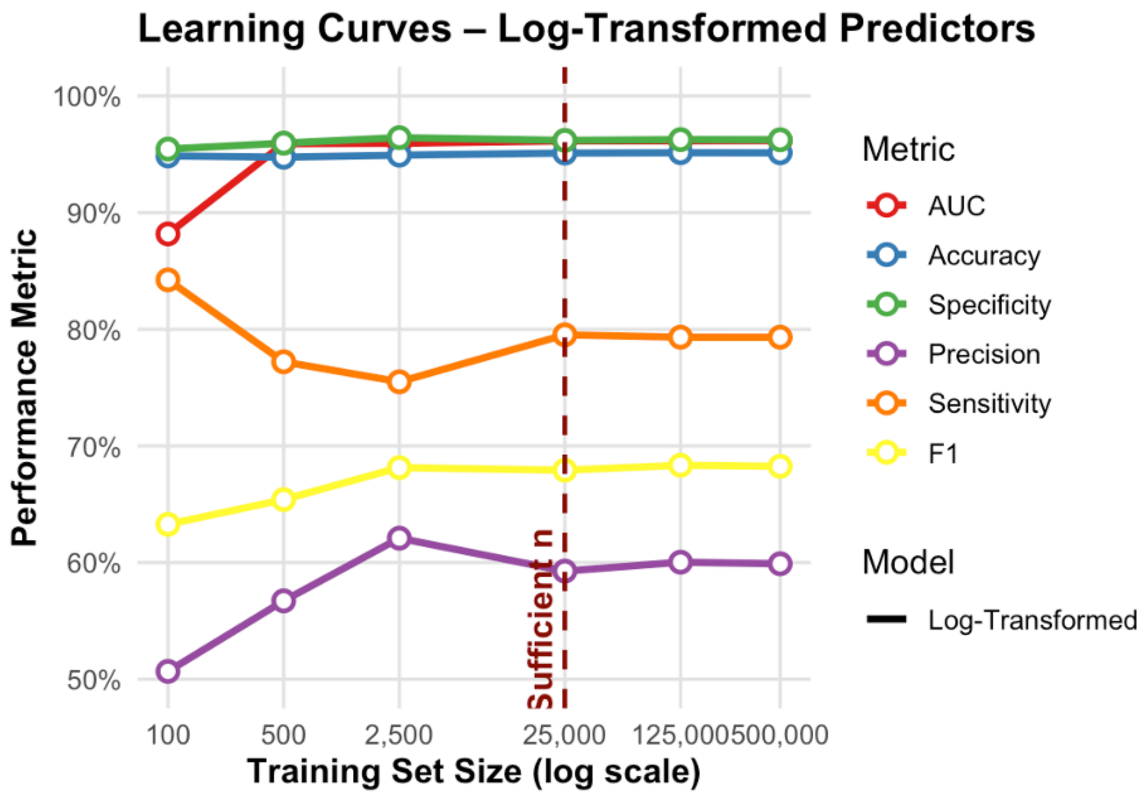
Now, we will validate our model assumptions. Our first concern is that the variables “distance from last transaction” and “distance from home” may be correlated. This would be an issue, as predictor variables should not be correlated with one another. To test for both a linear and a monotonic relationship, I used the Pearson and Spearman correlation tests. Both tests gave a near-zero relationship (.0002, -.0011 respectively). This indicates that the variables are not correlated (Kou et al., 2004).

We also had a concern that the binary variables “Used Chip” and “Used PIN” may be correlated. Since these are not continuous variables, we had to use a separate test called Cramér’s V. When utilizing this test, we received an output of .0014, where a result near 0 implies no correlation. Our concerns over multicollinearity were quelled (Kou et al., 2004).

Next, we also had to test if our predictor variables had a linear relationship with the log-odds of the response variable. Since this only applies to continuous variables, we only had to test “distance from home”, “distance from last transaction”, and “ratio to median purchase price”. In this test, a lambda of 1 would indicate no change is necessary, while other values correspond with other transformations. All 3 variables returned a lambda of 0, indicating that a log-transformation was necessary.

After identifying the issue, we transformed the 3 variables with a log (x+1) transformation (the +1 is to ensure none of our data has boundary issues), and re-ran our model. The test statistics are displayed below.

Figure 2:

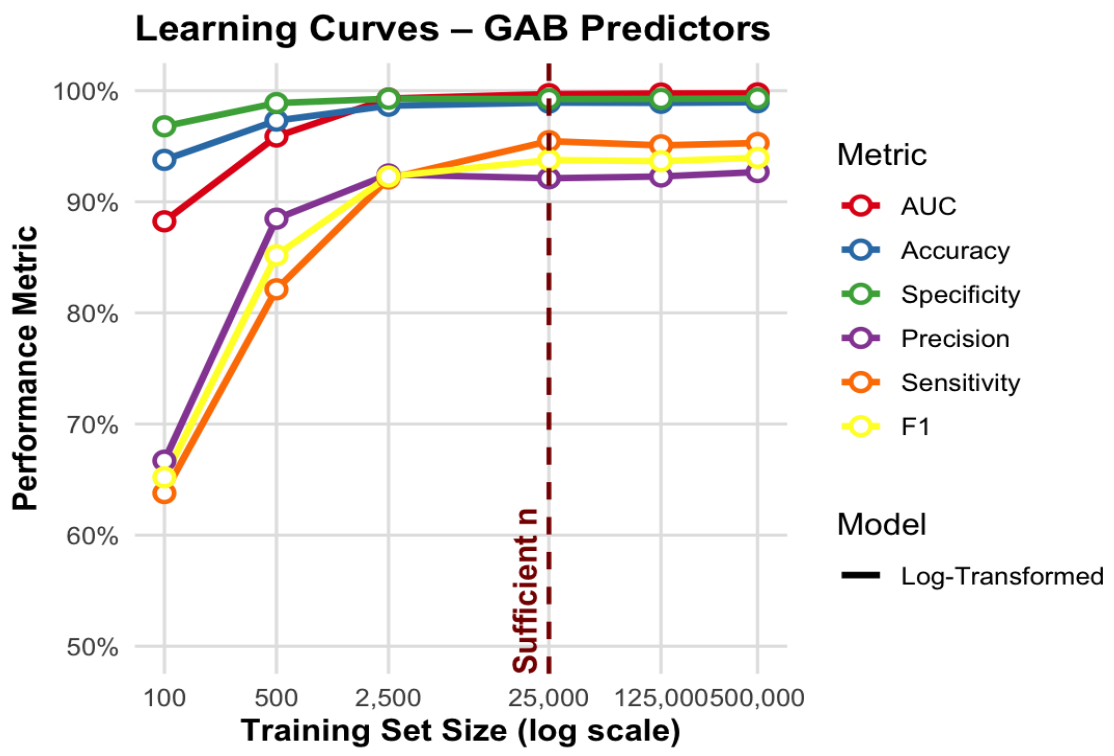


Note: The graph above shows the test statistics and how they change as a measure of sample size. Test statistics level out after an n of 25,000. The final test statistics using the full training set were as follows: AUC - 96.1%, Accuracy - 95.1%, Sensitivity - 79.3%, Precision - 59.9%, Specificity - 96.2%, F1 - 68.3%.

Despite making transformations to make our model validated, our test statistics went down across the board. The most impacted was sensitivity, which dropped over 10%. Typically, fixing a model to meet its assumptions would help its accuracy, but in this case it did not. We needed to test our model assumptions again. To do this, we used a standard Box-Tidwell test, which is used to test to see if the linearity in log-odds assumption is being met. When testing our transformed variables, the Box-Tidwell test still led us to conclude that this assumption was not being met. The best recourse for this issue is to use a new model.

GAMs are useful in this situation as they do not rely on the aforementioned assumption and are specifically designed to pick up on non-linear relationships (Bhattacharyya et al., 2011). For the GAM, we still decided to use the log-transformed variables (along with the binary variables) as predictors instead of the original data, as it still benefits the model to have less extreme outliers. The GAM test statistics are displayed below

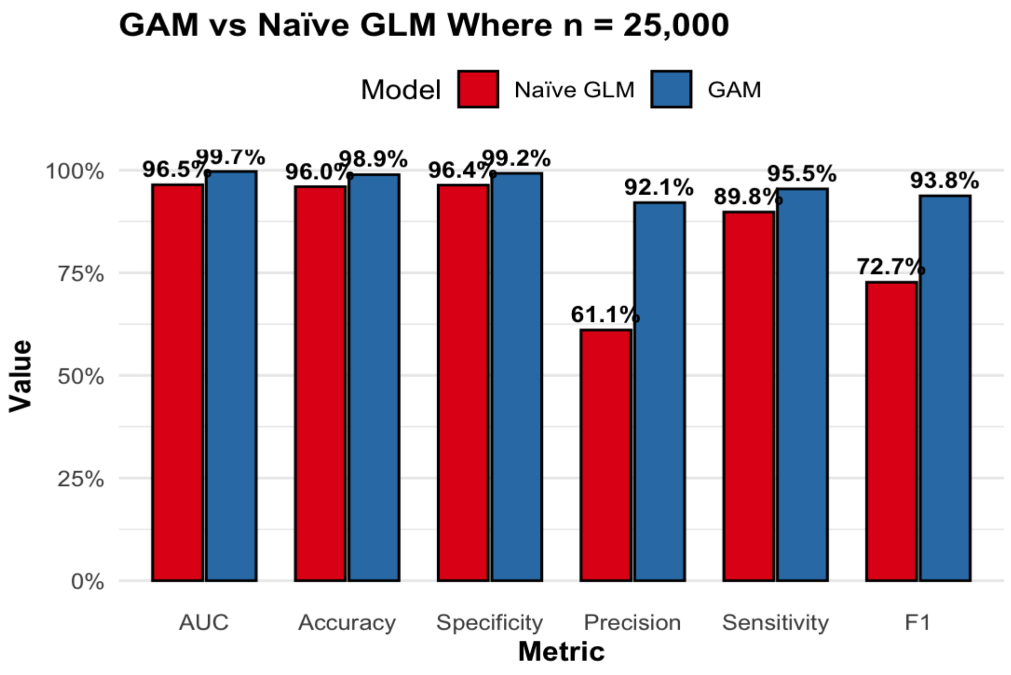
Figure 3:



Note: The graph above shows the test statistics and how they change as a measure of sample size. Test statistics level out after an n of 25,000. The final test statistics using the full training set were as follows: AUC - 99.8%, Accuracy - 99.0%, Sensitivity - 95.3%, Precision - 92.7%, Specificity - 99.3%, F1 - 94.0%.

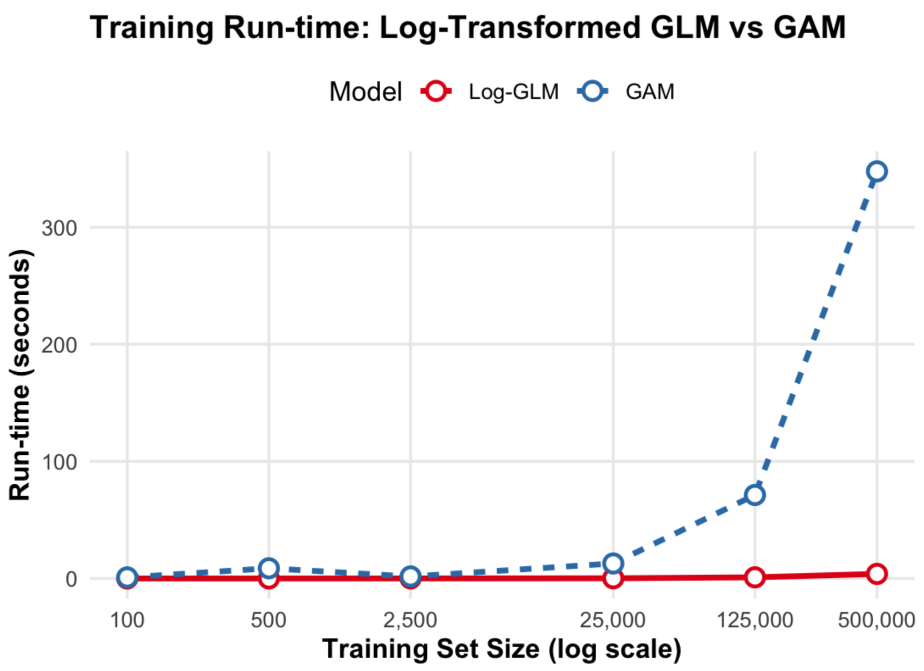
Using the GAM, we see that all of our test statistics increase. In figure 4 (below), we can see a direct comparison of the naive model and the GAM. The GAM model is better at both detecting fraudulent transactions and not misidentifying legitimate transactions. The model also levels out at 25,000 samples, giving us a borderline estimate on how large of a sample size is needed to create a sufficiently accurate model.

Figure 4:



While the GAM model is ostensibly better than our naive model, it is also important to note our other factor - run time. GAMs are much more computationally intensive, and increase superlinearly with sample size. The run time for both models is shown below:

Figure 5:



Note: The graph above shows the run time of GLMs and GAMs and how they increase as a measure of sample size. The logistic regression models run times were between 0 and 5 seconds, while the GLMs run time went from 12.8s at 25,000 to 72.3s at 125,000 and 351.6s at 500,000.

Figure 5 displays how GLMs maintain a small run time, even with large sample sizes, while the GAMs run time increased drastically with an increase in sample size. The run time was not a serious issue for our data, as we found that the GAM was sufficient at a sample size of 25,000 (where the run time was only 12.8s). Even at our maximum sample size, the run time was still under 6 minutes. However, this may become an issue when implementing with real life data, as with a smaller fraud chance, the models will become more imbalanced, requiring larger sample sizes to become sufficient. This reflects a well-documented issue in real-world credit card fraud detection, where fraudulent transactions are extremely rare relative to legitimate ones, often exceeding ratios of 10,000:1 (Sahin & Duman, 2011).

Overall, we were able to conclude that both GLM's and GAM's have sufficient sample sizes, where adding more data does not provide benefit to the model. Using this conclusion, we were able to identify ideal sample sizes for our simulated datasets, although it is to be noted that these sample sizes are not necessarily ideal for real datasets. We have examined the role where logistic regression models could be utilized for credit card fraud analysis, as they showed moderate success while remaining extremely fast to train, consistent with prior findings that logistic regression serves as a strong and efficient baseline despite being outperformed by more flexible machine learning models (Sahin & Duman, 2011; Bhattacharyya et al., 2011). Despite this, we still came to the conclusion that GAMs are to be preferred, as, despite how resource intensive the modeling is, they are much better equipped at detecting the non-linear relationships that appear in detecting credit card fraud. Finally, we suggest that further research should target applying GAMs to real-world fraud data, using industry-grade hardware that could cipher through much larger datasets with more variables.

Bibliography

Narayanan, D.R. (2022) *Credit Card Fraud*, *Kaggle*. Available at: https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud/data (Accessed: 30 September 2025).

John, G H, & Langley, P (1996). Static versus dynamic sampling for data mining. Available at https://cdn.aaai.org/KDD/1996/KDD96-069.pdf

Y. Sahin and E. Duman, "Detecting credit card fraud by ANN and logistic regression," *2011 International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, 2011, pp. 315-319, doi: 10.1109/INISTA.2011.5946108. Available at https://ieeexplore.ieee.org/abstract/document/5946108

B. Teh, M. B. Islam, N. Kumar, M. K. Islam and U. Eaganathan, "Statistical and Spending Behavior based Fraud Detection of Card-based Payment System," 2018 International Conference on Electrical Engineering and Informatics (ICELTICs), Banda Aceh, Indonesia, 2018, pp. 78-83, doi: 10.1109/ICELTICS.2018.8548878. Available at https://ieeexplore.ieee.org/abstract/document/8548878

Yufeng Kou, Chang-Tien Lu, S. Sirwongwattana and Yo-Ping Huang, "Survey of fraud detection techniques," IEEE International Conference on Networking, Sensing and Control, 2004, Taipei, Taiwan, 2004, pp. 749-754 Vol.2, doi: 10.1109/ICNSC.2004.1297040. Available at https://ieeexplore.ieee.org/abstract/document/1297040

Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, J. Christopher Westland, “Data mining for credit card fraud: A comparative study”,Decision Support Systems, Volume 50, Issue 3, 2011, pp. 602-613, https://doi.org/10.1016/j.dss.2010.08.008. Available at https://www.sciencedirect.com/science/article/pii/S0167923610001326