**Computational Statistics in Fraud Detection and Cybersecurity – Theoretical Writeup**

Cybersecurity holds paramount importance in the financial industry given the sensitivity of financial data and the magnitude of loss that can occur from data breaches or undetected fraudulent transactions. The financial industry relies increasingly on digital technologies, which leads to higher susceptibility to data threats and fraudulent transactions. Luckily, computational statistics techniques can provide an avenue for improving cybersecurity in this field by detecting fraudulent transactions, identifying stolen identities, flagging data anomalies to protect financial assets and properties. By investigating methods for fraud and anomaly detection, we will be able to provide valuable insights into how computational statistics can be used to mitigate threats and minimize loss in the financial industry.

Our project aims to employ Fisher Scoring, Monte Carlo Simulation, and Kernel Density Estimation for detecting fraudulent credit card transactions and conducting comparative analysis of model performance. Each team member will independently implement one of the three methods using a credit card dataset sourced from Kaggle. Upon completion and execution of all three models, we will assess their performance to identify the optimal model for predicting fraudulent credit card transactions. Finally, we will allocate majority of our effort towards the theoretical documentation of the best-performing model

Fisher Scoring achieved the highest accuracy rate of 96% during our cross-validation process among the three models we implemented for detecting fraudulent credit card transactions, so we will focus most of our theoretical documentation on this model.

Fisher Scoring is an iterative optimization algorithm used to find the maximum likelihood estimates of parameters or coefficients in various statistical models. It can be applied to logistic regression and generalized linear models to estimate regression coefficients through iterative numerical methods until the parameters reach convergence or steady state. Please see the step-by-step breakdown on how Fisher Scoring can be applied in logistic regression to estimate coefficient parameters.

**STEP 1.** Define logistic regression and probability function. Note for simplicity, only one x-variable is used in our equations below. In our data implementation using RStudio, we used multiple x-variables to model fraudulent credit card transactions. In logistic regression, the probability 𝑝 of an event occurring given predictor variable set is the following:

A math equations on a white background

Description automatically generated

**STEP 2.** Define likelihood function with L(β) represents the probability of event given parameter β. The likelihood function L(β) as defined as follows:

A black text on a white background

Description automatically generated

**STEP 3.** Define Log-Likelihood function by taking the natural log of the likelihood function to simplify the equation.

A group of math equations

Description automatically generated

**STEP 4.** Calculate the gradient of the Log-Likelihood function with respect to the coefficient.

A math equations on a white background

Description automatically generated

**STEP 5.** Calculate the Hessian matrix, which is a matrix of the second derivatives of the log-likelihood function with respect to the coefficients as follows:



**STEP 6.** Iteratively updates the coefficient vector 𝛽 using Fisher Scoring using the gradient and Hessian matrix calculation from previous steps to determine the direction and magnitude of each adjustment.



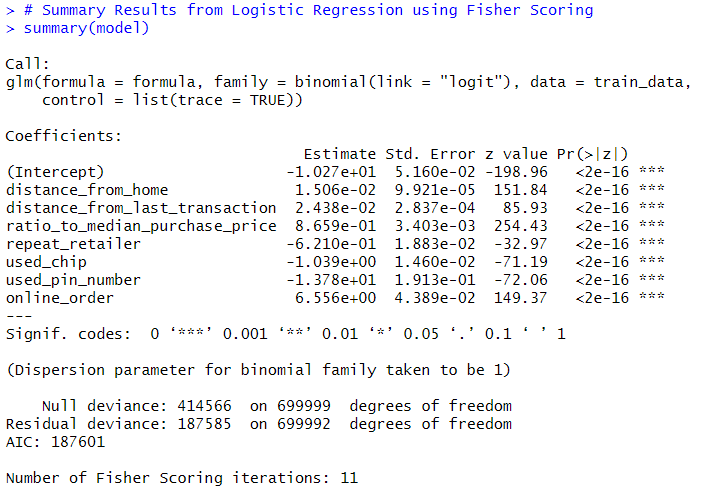
For this step, we also used the equations in the (Givens & Hoeting, 2013) textbook to guide our understanding and the formulation of the R-Code calculations, which we have included below for easy reference.

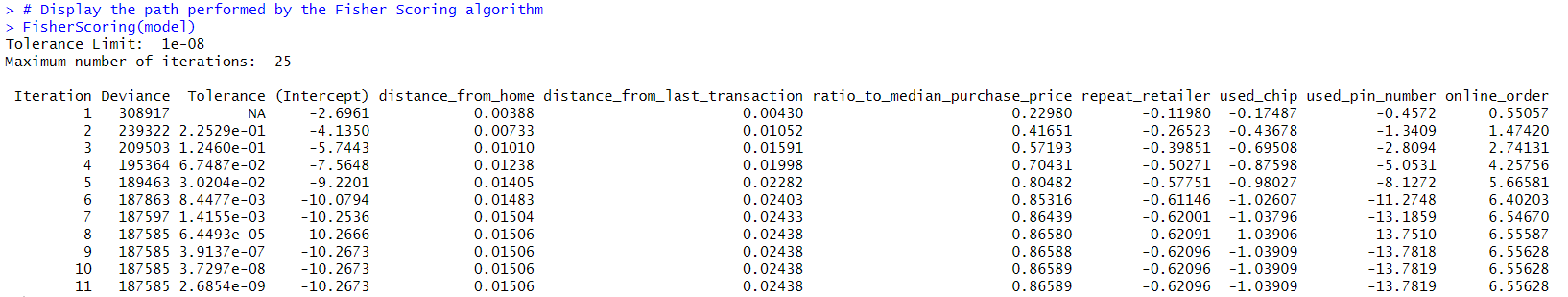
A white text with black and white text

Description automatically generated

**STEP 7.** Check for convergence by comparing the change in the log-likelihood or the coefficient between iterations.

Fortunately, we can perform these computational steps using RStudio and not by hand. Now, let us look at the application of mathematical theories in a real case example relating to cybersecurity in the financial industry, specifically relating to detecting fraudulent credit card activities. The Fisher.R code demonstrates the implementation of Fisher Scoring for detecting fraudulent credit card transactions using logistic regression with the specified parameters:  
formula <- as.formula("fraud ~ distance\_from\_home + distance\_from\_last\_transaction + ratio\_to\_median\_purchase\_price + repeat\_retailer + used\_chip + used\_pin\_number + online\_order").

The implementation used a dataset that contains one million non-missing credit card transactions with fraud as a binary response variable. First, the data is partitioned into training and testing datasets. Next, a logistic regression model using Fisher Scoring is fitted on the training dataset. Then, the trained model predicts “fraud” events on the testing dataset, and the performance score is calculated. We observed that by using Fisher Scoring, we can accurately predict fraudulent behavior 96% of the time. Statistical output shows the parameters of the logistic regression were adjusted 25 times before convergence, reaching steady states with the final coefficients or betas. Please refer to our calculation outputs below:   




The second-best performing model in predicting fraudulent credit card transactions is the Monte Carlo simulation, which generates random samples from a probability distribution to approximate the probability of outcomes that may be complex or unknown. The samples that are generated are from distributions that represent the uncertainty in a model. Once samples are generated, they are plugged back into the model and the results are compiled to analyze the behavior of the model. This method is helpful in situations where there is uncertainty in the model, and we want to understand its behavior. The code written reads the credit card fraud dataset, calculates the probability of fraud in the data, and then simulates new data with the Monte Carlo method. Then, it checks if fraud is detected or not, and calculates the overall detection rate of the simulation. This allows us to examine the effectiveness of the algorithm for detecting fraud, and we can see that it has a 95% detection rate, which demonstrates that it is also highly effective.

The third-best performing model is the Kernel Density Estimation, which is a non-parametric technique that can be utilized to estimate the probability density function of a random variable based on a sample of data points. This technique detects unusual patterns or outliers that deviate significantly from an expected distribution. Additionally, the KDE method can include a method to assign anomaly scores to data points and compare them to a certain threshold to make the system more robust in detecting fraud and anomalies. The kde.R code demonstrates how Kernel Density Estimation can be applied to detect fraud in credit card transactions. Using the linear discriminant analysis model with the parameters for “amount” and “newbalanceOrig” to determine the fraudulent transactions (see reference in statistical table below). First the data is split into a training and test set, then applying kernel density estimation is fit on the dataset. Then the linear discriminant analysis is trained with the respective data to determine the fraud events on the testing dataset. The performance score is calculated and displayed. We can observe that by using KDE, we are able to accurately predict fraudulent behavior 82% of the time. Additionally, please refer to our calculations below:

A screenshot of a computer

Description automatically generated

In summary, we can see that all three computational methods offer effective solutions to detect credit card fraudulent, with Fisher Scoring performing the best, Monte Carlo simulation ranking second, and Kernel Density Estimation as the third-best model. For this exercise, we performed the models independently to assess model performance, but real-world financial industries often use a combination of models to optimize accuracy in fraud detections. Our implementations demonstrate how computational statistics play a pivotal role in strengthening cybersecurity within the financial industry. By leveraging these computational statistics techniques, financial institutions can mitigate the risk of fraudulent transactions, stolen identity, and cyber-attacks to protect sensitive and valuable digital assets for the great good.