McGill University

MGSC 416: Data-driven Models for Operations Analytics

Professor Rim Hariss

13 April 2023

Final Project Report

Detecting Credit Card Fraud

Group 9

Tina Dimova (260986550)

Yuxin Gu (260954127)

Cherie Koh (261132411)

Table of Contents

1.0 Introduction

1.1 Background

With the increasing number of digital payments, cybercriminals are also evolving. Detecting fraud in the trillions of daily credit card transactions is becoming increasingly challenging. In 2020, credit card fraud losses amounted to an estimated US$28.58 billion, and this figure is expected to rise to US$49.32 billion by 2030 (Nilson Report, 2021). A study conducted by KPMG found that more than 50 percent of financial institutions surveyed recover less than 25 percent of their fraud losses, emphasizing the critical importance of fraud prevention (KPMG, 2019). Thus, financial institutions must be able to respond quickly to security breaches and adopt technologies to detect and prevent fraud. Efficient and accurate fraud detection systems are crucial to detect fraud.

1.2 Problem Definition and Formulation

Our project aims to build two models. The first model is a predictive model developed using Python that can differentiate between fraudulent and non-fraudulent credit card transactions based on specific predictor variables. The second model aims to determine the appropriate threshold for detecting fraud by minimizing the squared errors of the first model using Python's optimization solver, Gurobi. This approach will help limit losses from undetected fraudulent transactions (false negatives) while ensuring customer satisfaction by avoiding unnecessary card suspensions from non-fraudulent transactions that are wrongly flagged by the system (false positives).

2.0 Data

2.1 Dataset

Our data for the prediction and optimization models are based on a credit card fraud dataset retrieved from Kaggle (Narayanan, 2022). The dataset contains eight variables, including one binary dependent variable, *fraud*, which determines whether a transaction is fraudulent (1) or not (0). The seven independent variables consist of three continuous variables and four binary variables:

- *distance_from_home*: the transaction's distance from the cardholder's home
- *distance_from_last_transaction*: the transaction's distance from the cardholder's last transaction

- *ratio_to_median_purchase_price*: ratio of the transaction's price to the cardholder's median purchase price
- *repeat_retailer*: (binary) determines whether the transaction happened from the same retailer as the previous one
- *used_chip*: (binary) whether the transaction is done through a chip (credit card)
- *used_pin_number*: (binary) whether the transaction is done through PIN
- *online_order*: (binary) whether the transaction is an online order

2.2 Data Pre-processing

Upon evaluating our dataset, we observed that it originally had a 90-10 percent split between non-fraudulent and fraudulent transactions. However, the results derived from the original non-fraudulent and fraudulent transaction ratio prompted us to create a balanced sample of 10,000 data points with a 50-50 split would enhance the performance of the prediction and optimization model (refer to the prediction model for further details). This approach enables the model to gain more insight and knowledge from a larger dataset of fraudulent transactions, thereby improving its overall accuracy.

During the exploratory data analysis process, we noticed that the distribution of our three continuous independent variables (refer to Appendix A) exhibited a skewness. To address this, we performed log transformations on the variables, which resulted in a normal distribution that better aligns with the modelling assumptions. Subsequently, these log-transformed variables were merged with the discrete independent and dependent variables, forming the data frame to be used for the prediction and optimization model.

3.0 Methods

3.1 Prediction Model

We developed three prediction models - logistic regression, K-Nearest Neighbor (KNN), and Gradient Boosting Tree (GBT). To ensure comparability across models, we split the data with a test size of 0.33. For KNN, we standardized the dependent variables (X), while for logistic regression and GBT, we did not standardize the data as these models are insensitive to variable magnitude.

Initially, the logistic regression (unbalanced) model yielded the lowest accuracy and F1 score, while GBT produced the most promising results among the three models (as presented in Appendix B). From an accuracy standpoint, all three models produced an accuracy higher than 95%. However, upon analyzing the confusion matrix of the logistic regression model, we realized that only a small percentage of fraudulent transactions were accurately predicted (approximately 50%). This can be caused by the limited number of fraudulent transactions in the original dataset (277 out of 3300, approximately 8.39%), resulting in an unrealistically high accuracy score for all three models.

To address this issue, we created a balanced sample of transactions by randomly extracting 5000 data points for each category. As presented in Appendix C, the accuracy score of the logistic regression model decreased by approximately 10% to 0.8539. However, the F1 score and confusion matrix improved significantly (originally 63.49%), with approximately 87.67% of fraudulent transactions accurately predicted, which further validates our hypothesis that the original high accuracy and low F1 score are the results of an unbalanced dataset. In addition, KNN and GBT saw a slight decrease in accuracy (0.9806 and 0.9979, respectively), but both models exhibited an improvement in F1 score (0.9807 and 0.9979, respectively). Notably, GBT correctly predicted 100% of fraudulent transactions, significantly improving from the previous logistic regression results.

3.2 Optimization Model

Decision Variables

Our optimization model has the following decision variables:

- z = continuous; threshold at which a transaction is flagged for fraud
- y_{ip} = binary; whether the transaction is flagged for fraud based on this model

The following constant variables:

- y_i= binary; whether a transaction is fraudulent or not (truth)
- p_i= probability calculation of fraud based on logistic regression coefficients

  Note: The calculations of p_i (Appendix D) require the knowledge of the prediction model coefficients. We will utilize the logistic regression coefficients as they offer more interpretability than GBT

And the following auxiliary variables:

- $d_{i,j}$= binary; whether a cost is selected
- c_i= continuous; cost of errors (false positive = \$1, false negative =\$2)

Objective Function

The objective function seeks to minimize the cost of errors while finding a new optimal fraud threshold "z".

$$MIN \sum c_i (y_{ip}\text{-}y_i)$$

Constraints (All numerical constraints can be found in Appendix E)

Threshold Constraints

The threshold in our optimization model is a value between 0 and 1. If the calculated logistic probability for a given transaction (P_i) is greater than or equal to the threshold value (z), the model assigns a value of 1 (fraudulent) to y_{ip}. Otherwise, the model assigns a value of 0 (non-fraudulent) to y_{ip}.

Cost Constraints

Once the predicted outcome (y_{ip}) is determined, the model incurs a cost penalty of \$1 if y_{ip} is greater than the true value (y_i), indicating a false positive (predicted 1, actual 0). This cost reflects the negative impact of cardholder dissatisfaction resulting from the falsely blocked card. Conversely, if y_{ip} is less than the true value (y_i), indicating a false negative (predicted 0, actual 1), a cost penalty of \$2 is assigned to reflect the negative impact of monetary losses due to undetected fraud. No penalty (\$0) will be given if the model correctly predicted the transaction ($y_{ip} = y_i$).

No Non-negativity Constraints

Note that non-negativity constraints are not required for our optimization model since most of the variables are coded as binary, meaning that they can only take on the values 0 or 1. Additionally, the two continuous variables we have are already constrained within their respective ranges: z is between 0 and 1, and ci can be negative, as some costs have negative coefficients.

Rationale Behind Selected Costs for Errors

To determine the costs to assign to the two types of errors, we experimented with 5 different variations of the ratio of cost for a false positive and cost for a false negative (see Appendix F for more details). The notable insights from the 5 attempts are summarized in Appendix G

After conducting a thorough analysis of the model's performance, we found a trade-off between accuracy and recall, a common challenge in data modelling. ***Accuracy score*** is the number of correct predictions the model makes divided by the total number of predictions made. ***Recall score*** is the true positive rate calculated as True Positives/(True Positives+False Negatives). In this particular model, we prioritize recall since the objective is to minimize the costly errors resulting from undetected fraudulent transactions.

When different cost ratios are applied to the model, we observed that increasing the cost of a false negative while keeping the cost of false positive constant leads to an increase in recall but a decrease in accuracy. While financial institutions aim to improve their ability to correctly predict fraudulent transactions due to the actual threat of losing money, it is still important to select costs that will not result in too significant an increase in false positives, as this can still result in customer dissatisfaction.

Furthermore, we discovered diminishing marginal returns regarding model improvement as the cost for false negatives increases while that of a false positive stays constant. For example, moving from a cost ratio of 1:1 to 1:2, we observed a significant 10% increase in recall and a decrease in false negatives of 467 cases. However, moving from 1:2 to 1:3 only resulted in a marginal 2% increase in recall and a decrease in false negatives of 99 cases. We also evaluated the results for a ratio of 1:10, where the number of false negatives decreased to 52, and the recall score increased to 98.96%. However, the number of false positives significantly increased from 1576 in the 1:3 ratio to 2768 cases in the 1:10 ratio, resulting in an accuracy of 71.8%. For this reason, we deemed this tradeoff less desirable even though it had the best recall.

Considering these findings, we decided to implement the error costs of $1 for a false positive and $2 for a false negative to strike a balance between maximizing recall and minimizing the number of false positives. This decision was reached after carefully considering the trade-offs between accuracy and recall, the potential consequences of false negatives and false positives, and the cost implications for financial institutions.

4.0 Results

In terms of prediction, it is clear that Gradient Boosting Tree has the best performance with an accuracy score of 0.9979 and recall of 1 indicating an ability to correctly predict all fraudulent transactions, which is the aim of our research. However, GBT lacks the interpretability given by logistic regression. Through our optimization, we were able to improve the logistic regression by playing with the threshold value and found that the optimal threshold is 0.3999 with consideration of our constraints and cost penalty ratio. This is lower than the default logistic regression threshold of 0.5 implying that the model is more strict towards fraudulent transactions and is able to capture more fraudulent transactions at the expense of false positive errors. This explains why the accuracy of this new model (Appendix H) took a slight dip from 0.8539 to 0.8300 while the recall increased from 0.8767 to 0.9198.

5.0 Limitations and Next Steps

One significant limitation of our models is the lack of validity of the dataset used. As the project progressed, we discovered that the source of the transaction data was undisclosed, leaving us unable to verify whether the data was real or fabricated. Furthermore, the dataset did not include critical metrics such as unique customer IDs, customer profile information (such as gender and age), and transaction amounts. Incorporating customer identification and profiling would be interesting in our prediction model, as these attributes are known to influence the likelihood of fraud. For example, a senior citizen may be more vulnerable to fraud schemes than a tech-savvy young adult. Moreover, including transaction amounts in our optimization model would enable us to quantify the actual financial losses resulting from undetected fraud.

In practice, GBT has generally been considered the optimal algorithm for fraud prediction. However, for financial institutions that value model interpretability and seek to gain a better understanding of how various variables impact the outcome of fraud, our report demonstrates that logistic regression can still be employed while improving its threshold to minimize losses - both explicit (financial) and implicit (customer satisfaction).

6.0 References

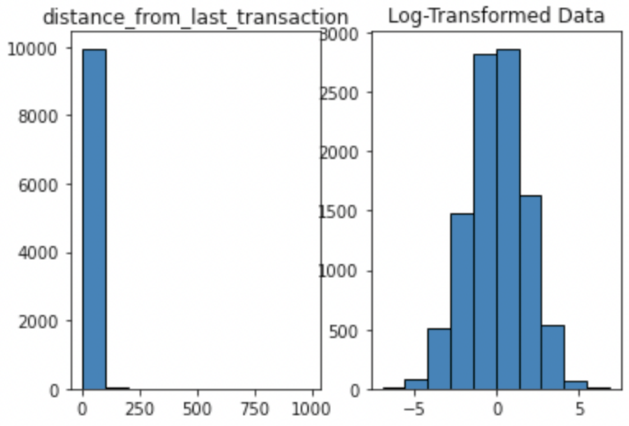
KPMG. (2019, May 16). *The multi-faceted threat of fraud*. Retrieved April 3, 2023, from https://kpmg.com/xx/en/home/insights/2019/05/the-multi-faceted-threat-of-fraud-are-banks-up-to-the-challenge-fs.html

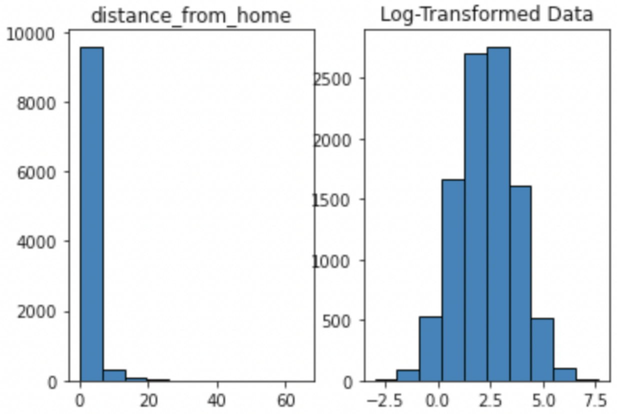
Narayanan, D. (2022). Credit Card Fraud. Retrieved April 3, 2023 from https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud

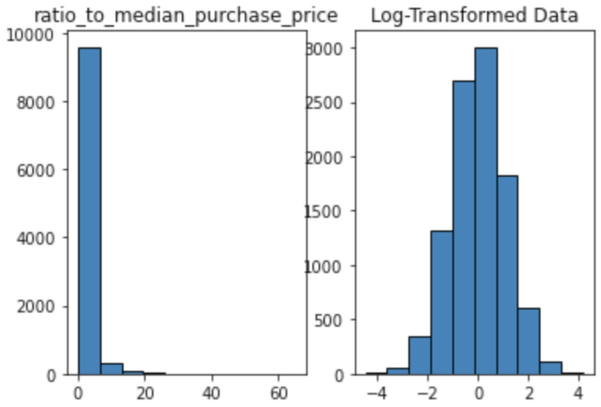
Nilson Report. (2021). *Nilson Report* (Issue 1209). Retrieved April 3, 2023, from https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf

7.0 Appendices

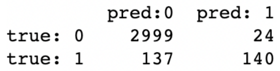
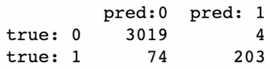
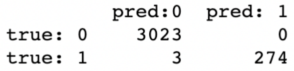
Appendix A: Log Transformation of Continuous Independent Variables



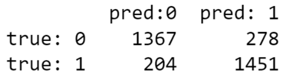
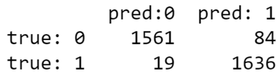
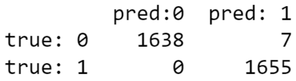




Appendix B: Initial Prediction Model Accuracy Metrics

	Logistic Regression	K-Nearest Neighbor	Gradient Boosting Tree
Accuracy Score	0.9512	0.9797	0.9991
F1-Score	0.6349	0.9011	0.9946
Precision	0.8537	0.9807	1.0
Recall	0.5054	0.7329	0.9892
Confusion Matrix	 pred:0 pred: 1 true: 0 2999 24 true: 1 137 140	 pred:0 pred: 1 true: 0 3019 4 true: 1 74 203	 pred:0 pred: 1 true: 0 3023 0 true: 1 3 274

Appendix C: Prediction Model Accuracy Metrics (50/50 split)

	Logistic Regression	K-Nearest Neighbor	Gradient Boosting Tree
Accuracy Score	0.8539	0.9806	0.9979
F1-Score	0.8576	0.9807	0.9979
Precision	0.8392	0.9512	0.9958
Recall	0.8767	0.9885	1.0
Confusion Matrix	 pred:0 pred: 1 true: 0 1367 278 true: 1 204 1451	 pred:0 pred: 1 true: 0 1561 84 true: 1 19 1636	 pred:0 pred: 1 true: 0 1638 7 true: 1 0 1655

Appendix D: Calculating the logistic probability function

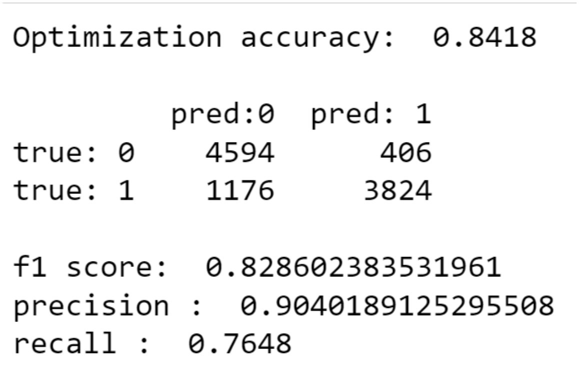
$$p = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n}}$$

Appendix E: Optimization Model Constraints

Constraint	Explanation
$0 \leq z \leq 1$	Threshold is between 0 and 1
$Y_{ip} \leq p_i - z + 1$	When $P_i \geq z \longrightarrow Y_{ip} = 1$
$Y_{ip} \geq p_i - z$	When $P_i < z \longrightarrow Y_{ip} = 0$
$c_i = 1{*}d_{i,0} - 3{*}d_{i,1} + 0{*}d_{i,2}$	Cost is determined by the $d_{i,j}$ auxiliary ● $d_{i,0}$: (FP) costs 1\$ ● $d_{i,1}$: (FN) costs 3\$ ● $d_{i,2}$: (TP or TN) costs 0\$ (correct prediction has no cost)
$d_{i,j} \leq y_{ip} - y_i + 1$	When $y_{ip} > y_i \longrightarrow d_{i,0} = 1$ (FP)
$d_{i,j} \geq y_{ip} - y_i$	When $y_{ip} < y_i \longrightarrow d_{i,1} = 1$ (FN)
$\sum d_{i,j} = 1$	Only 1 $d_{i,j}$ can be true

Appendix F: Cost Selection for Errors Based on Penalty Ratios (FP:FN)

F.1 Penalty ratio **1:0.5**

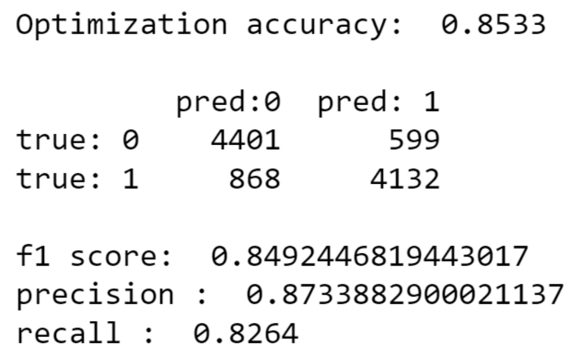


```
Optimization accuracy:  0.8418

          pred:0  pred: 1
true: 0    4594      406
true: 1    1176     3824

f1 score:  0.828602383531961
precision :  0.9040189125295508
recall :  0.7648
```

F.2 Penalty ratio **1:1**

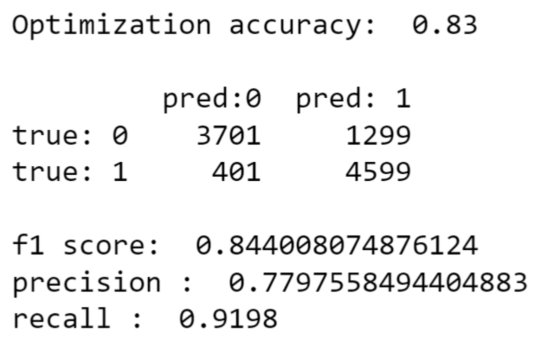


```
Optimization accuracy:  0.8533

          pred:0  pred: 1
true: 0    4401      599
true: 1     868     4132

f1 score:  0.8492446819443017
precision :  0.8733882900021137
recall :  0.8264
```

F.3 Penalty ratio **1:2** (chosen)

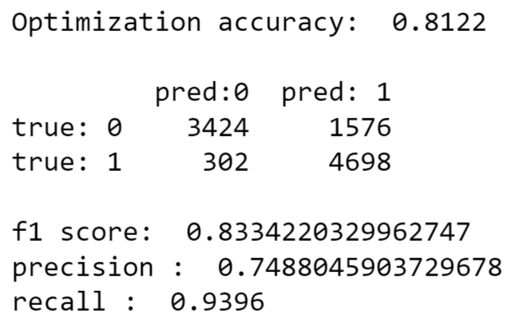


```
Optimization accuracy:  0.83

         pred:0  pred: 1
true: 0    3701     1299
true: 1     401     4599

f1 score:  0.844008074876124
precision :  0.7797558494404883
recall :  0.9198
```

F.4 Penalty ratio **1:3**

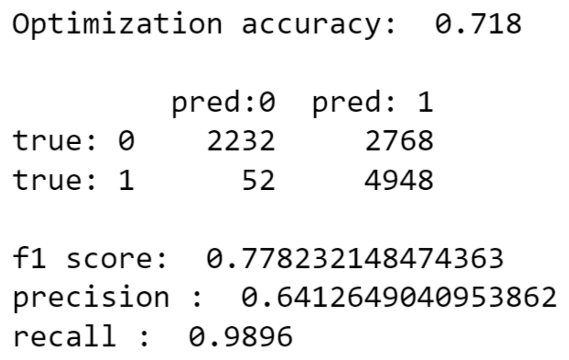


```
Optimization accuracy:  0.8122

         pred:0  pred: 1
true: 0    3424     1576
true: 1     302     4698

f1 score:  0.8334220329962747
precision :  0.7488045903729678
recall :  0.9396
```

F.5 Penalty ratio **1:10**



```
Optimization accuracy:  0.718

         pred:0  pred: 1
true: 0    2232     2768
true: 1      52     4948

f1 score:  0.778232148474363
precision :  0.6412649040953862
recall :  0.9896
```

Appendix G: Ratio Summary Table

Cost of FP	Cost of FN	Accuracy	Recall	No. of FP	No. of FN
1	0.5	0.8418	0.7648	406	1176
1	1	0.8533	0.8264	599	868
1	2	0.8300	0.9198	1299	401
1	3	0.8122	0.9396	1576	302
1	10	0.7180	0.9896	2768	52

Appendix H: Optimization results

Cost of FP	Cost of FN	Accuracy	Recall	No. of FP	No. of FN
1	2	0.8300	0.9198	1299	401