Project Scope Statement

CREDIT CARD FRAUD DATASET

Eric Lim, Klaas van Kempen, Jude Moukarzel
GEORGETOWN UNIVERSITY |

V1, rev 2023-09-07

Table of Contents

1.1 Project #	*1.2 Project Description*	*1.3 Date Submitted*	*1.4 Project Priority*
1	GT Bank is a nationwide bank located in Washington D.C. Every day, thousands of credit card transactions flow through their systems, and they would like to predict which ones are fraudulent. We are commissioned to build a machine learning application that the bank can use to flag all fraudulent transactions so that they can take action and protect their customers.	Sep. 9, 2023	Priority 01

1.5 Step 1. Project Deliverables

Please list *all project deliverables* listed in the Project Charter and, if necessary, elaborate on them. *Do not list dates*. Add more rows as necessary.

Deliverable ID#	**Description**
1	Problem Identification
2	Method Evaluations
3	Solution by Hyperparameter Optimization
4	Deployment Pipeline/Platform
5	Project Demo In class

1.6 Step 2. List of Project Tasks

Please list ***all project tasks*** to be completed, based on the "Deliverables" specified in the Project Charter. *Do not list dates*. Add more rows as necessary. Optional: you may substitute a work breakdown structure (WBS) or mind-map in lieu of Step 2. Please attach WBS or mind-map to the document.

Task ID#	**Task to be completed**	**Delivery Date**	**For Deliverable #**
1	Submit Project Charter / Problem Identification	09/07/2023	1
2	Submit Method Evaluations	09/28/2023	2
3	Submit Solution by Hyperparameter Optimization	10/12/2023	3
4	Submit Deployment Pipeline/Platform	10/26/2023	4
5	Project Demo in Class	11/16/2023	5

1.7 Step 3. Out of Scope

This project **will NOT accomplish or include** the following:	This project will not include any post-detection human decision making steps taken by the bank to deal with the fraudulent transactions identified. In other words, the scope of this project is solely on the detection of potentially fraudulent transactions but not the human intervention of actually validating the detection thereafter.

1.8 Step 4. Project Assumptions

Please list any project factors that will be considered to be true, real, or certain. Assumptions generally involve a certain degree of risk.

#	Assumption
1	The deployed model will be integrated with the bank's credit card systems so daily transactions can be picked up and analyzed real time.

1.9 Step 5. Project Constraints

Project Start Date	09/4/2023
Launch/Go-Live Date	12/05/2023
Project End Date	12/05/2023
List any hard deadline(s)	09/17/2023 09/28/2023 10/12/2023 10/26/2023 11/16/2023
List other dates/descriptions of key milestones	None
Budget constraints Enter information about project budget limitations	Total (maximum) project budget Limited funding for Cloud computing
Quality or performance constraints Enter any other requirements for the functionality, performance, or quality of the project	Software must load in 10 seconds or less; System must provide 99.9% uptime; Machine Learning algorithm must be trained on local machine and on subset of dataset due to no access to Cloud technology; Fraudulent transactions detected must be accurate at 90% accuracy or more.
Equipment/personnel Constraints Enter any constraints regarding equipment or people that will impact the project	Cloud technology won't be available until December 2023; Local machines are used to train/test the ML model; Employees Eric, Klaas & Jude are the only employees available to complete this work.
Regulatory constraints Enter any legal, policy or other regulatory constraints	Website must comply with CSU accessibility policy; Database must comply with campus Information Security policy.
Dataset Constraints	Missing data elements such as: ● Location of transaction ● Location of Customer ● Time between transactions (which may be resolved with feature engineering using the customer ID)

1.10 Step 6. Updated Estimates

Estimate T&C hours required to complete project	N/A	If charge-back project, list total estimated T&C cost	N/A

1.11 Step 7. Approvals

Required For Project Class…	Role of Approver	Submitted for Approval on:	Approval Received on:
All classes	1. Client + Client Supervisor	Nakul R. Padalkar	
All classes	2. T&C Supervising Manager	Nakul R. Padalkar	
Class 3 + 4 only	4. VP for Technology & Communication	Nakul R. Padalkar	
Class 3 + 4 only	5. Project Review Board	Nakul R. Padalkar	

2 Introduction

Credit card fraud in the USA is a widespread and persistent issue that involves the unauthorized use of credit or debit card information for financial gain. It occurs when individuals or criminal organizations obtain and use someone else's credit card details without their consent. Fraudulent transactions in the United States represent a pervasive and multifaceted challenge that affects individuals, businesses, and the economy at large.

Combating credit card fraud is an ongoing effort that requires the collaboration of various stakeholders and the adoption of advanced security measures. Some key strategies to combat fraud include but are not limited to:

1. secure payment technologies with two-factor authentication,
2. regular monitoring,
3. education,
4. customer verification and awareness, and
5. fraud detection systems.

This project delves into the use of Machine Learning algorithms to deploy a new fraud detection system that would analyze daily transaction data and identify any unusual patterns that could lead to potential fraud.

2.1 Project Background and Description

Fraudulent transactions can have significant and far-reaching impacts on various stakeholders, including individuals, businesses, financial institutions, and the broader economy. These impacts can be both immediate and long-term, such as financial loss, negative customer experience, impact on credit, economic damage, and much more.

The focus of this project is on the financial industry. Credit cards and mobile payments are very common and will continue to become more integrated in day-to-day transactions. Therefore, it is imperative that banks find a way to address this issue in a seamless manner and ensure that the customer is always left satisfied in the case of a fraudulent transaction. For the purposes of both user security and customer experience, banks need to be able to catch fraud when it actually occurs and quickly remediate false positives. Mitigating financial loss while minimizing disruptions must be a priority.

2.2 Proposed Methods

In order to deploy our fraud detection application, the team will be evaluating several Machine Learning algorithms using various ensemble methods, both generative and non-generative, in a Binary Classification exercise to detect whether or not a transaction is fraudulent.

Generative methods include boosting algorithms such as AdaBoost, Gradient Boost, and XGBoost, while non-generative methods include bagging, random forests, and stacking. Hyperparameter tuning will then be performed for all methods in order to optimize the fraud detecting algorithm and find the highest performing model with maximum accuracy, precision, and recall, as well as the most important features. The algorithm will then be deployed on an application and delivered to the bank.

3 Analysis of the Dataset and Trained Model

3.1 Exploratory Analysis and Visualization

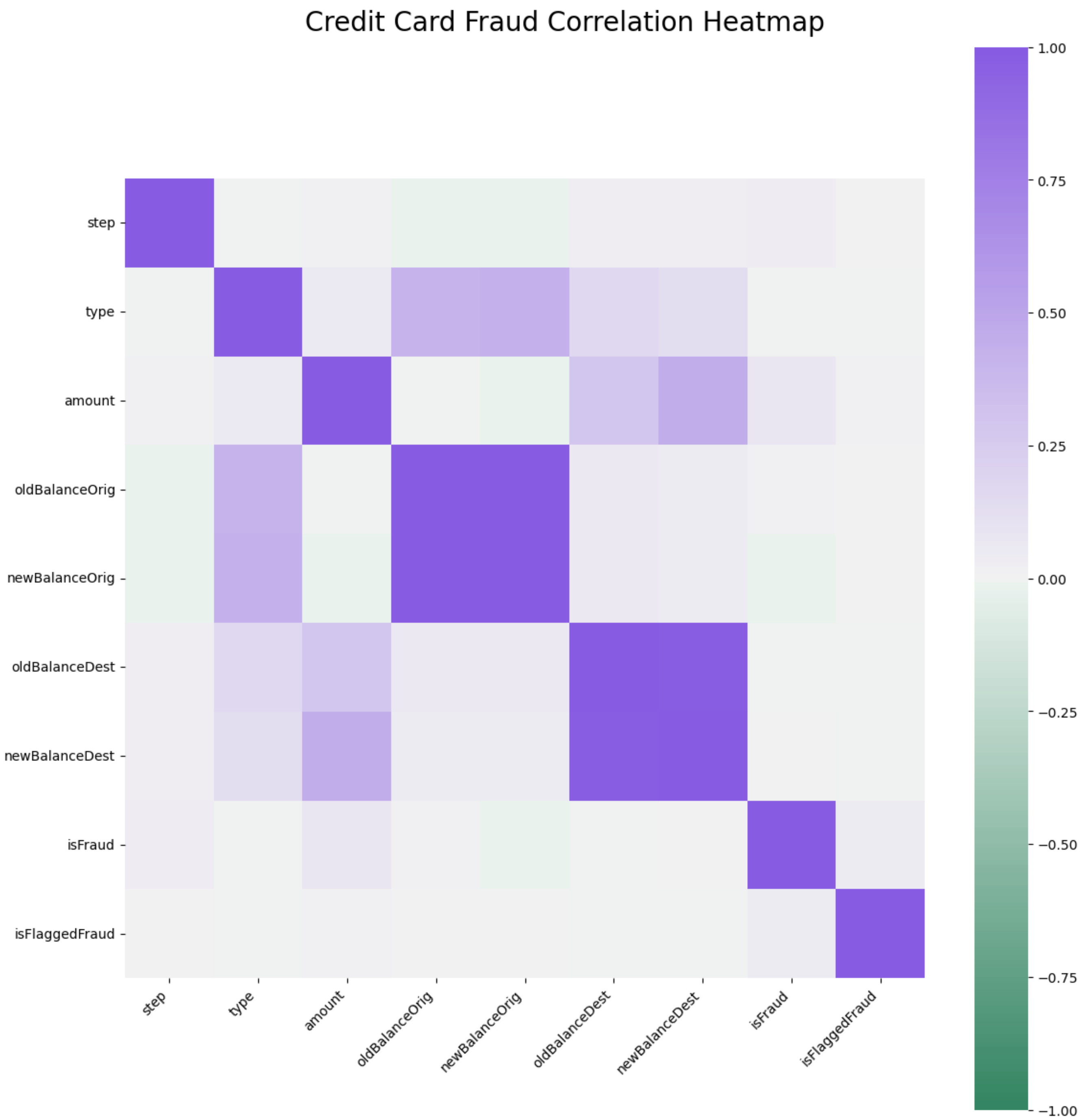


Figure 1: Correlation Heatmap of the numeric features in the credit card fraud dataset. There appears to be some multicollinearity between logically connected independent variables. For example, `oldBalanceOrig` and `newBalanceOrig` almost linearly correlate, as well as `oldBalanceDest` and `newBalanceDest`. Overall, however, features that aren't logically related appear to be independent of each other.

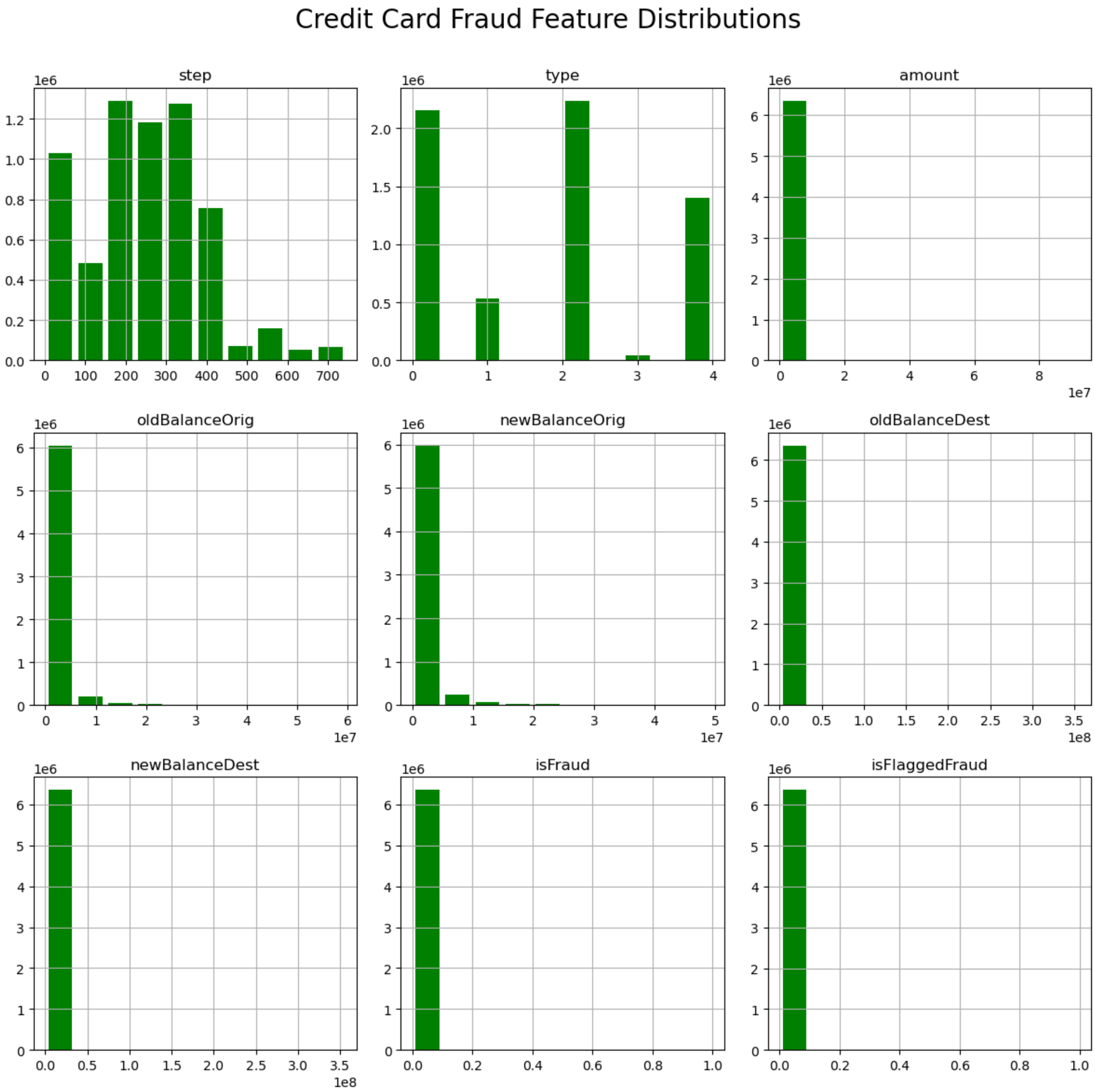
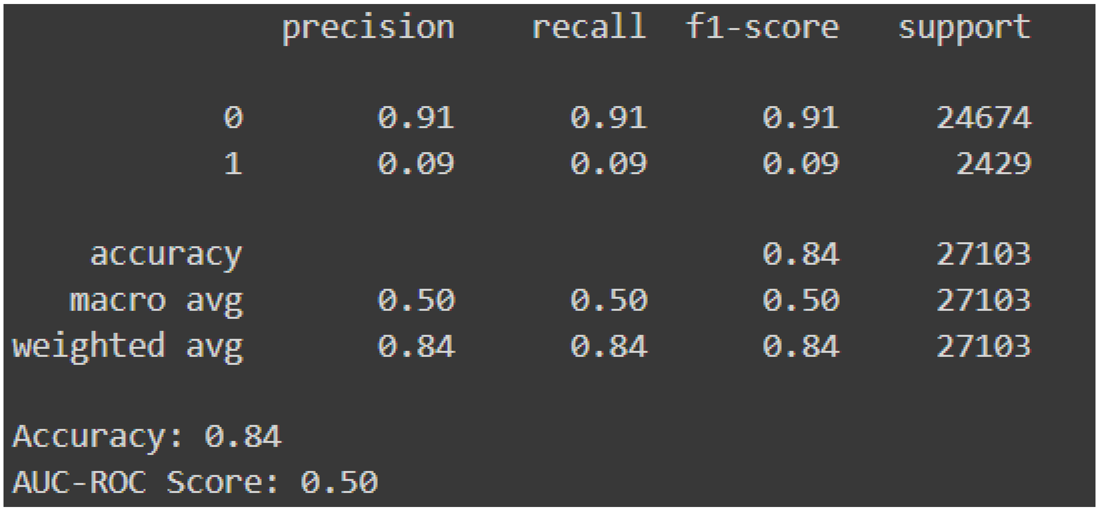


Figure 2: Feature Distributions of the numeric features in the credit card fraud dataset. Features representing the transactions (`amount`, `oldBalanceOrig`, `newBalanceOrig`, `oldBalanceDest`, `newBalanceDest`) are heavily right skewed, which coincides with a general understanding of income and wealth distribution (i.e. Pareto's Law of Income Distribution). The target variable (`isFraud`) shows that the dataset is heavily imbalanced, with over 6 million observations of non-fraud and only 8,000 observations of fraud.

3.2 Baseline Model

Since this is a classification dataset, we have decided to analyze the model using the scikit-learn's DummyClassifier using a stratified strategy to account for the heavily imbalanced dataset. We will also assume this to be the baseline model and improve the performance by using multiple ensemble techniques. For simplicity, all the model code (and code for the rest of the analysis) is available with an attached jupyter notebook and will not be included in the report. It should also be noted that we have performed random downsampling of the majority class to make computation times more reasonable. Rather than having about 6 million non-fraud and 8,000 fraud rows, we created a new dataset with all of the fraud rows (8,213 rows) and 82,130 non-fraud rows (i.e. a 10:1 ratio of non-fraud to fraud rows).



```
              precision    recall  f1-score   support

           0       0.91      0.91      0.91     24674
           1       0.09      0.09      0.09      2429

    accuracy                           0.84     27103
   macro avg       0.50      0.50      0.50     27103
weighted avg       0.84      0.84      0.84     27103

Accuracy: 0.84
AUC-ROC Score: 0.50
```

Figure 3: Model Metrics for the DummyClassifier (Baseline Model): The Baseline Model, due to an imbalanced dataset, shows poor performance.

The baseline classifier has an accuracy of 84%. Although it may seem reasonable at a glance, it performs worse than indiscriminately choosing the majority class, which would net an accuracy of about 91% (since a 10:1 ratio would lead to 10/11 ⇒ 0.9090). We can also see poor performance in the precision, recall, and f1-score, where there's a stark contrast between the majority and minority class. Lastly, the AUC-ROC score, which would be a better metric in interpreting the performance of a model on a heavily imbalanced dataset, is 0.5, which suggests that the model has no ability to discriminate between the two classes.

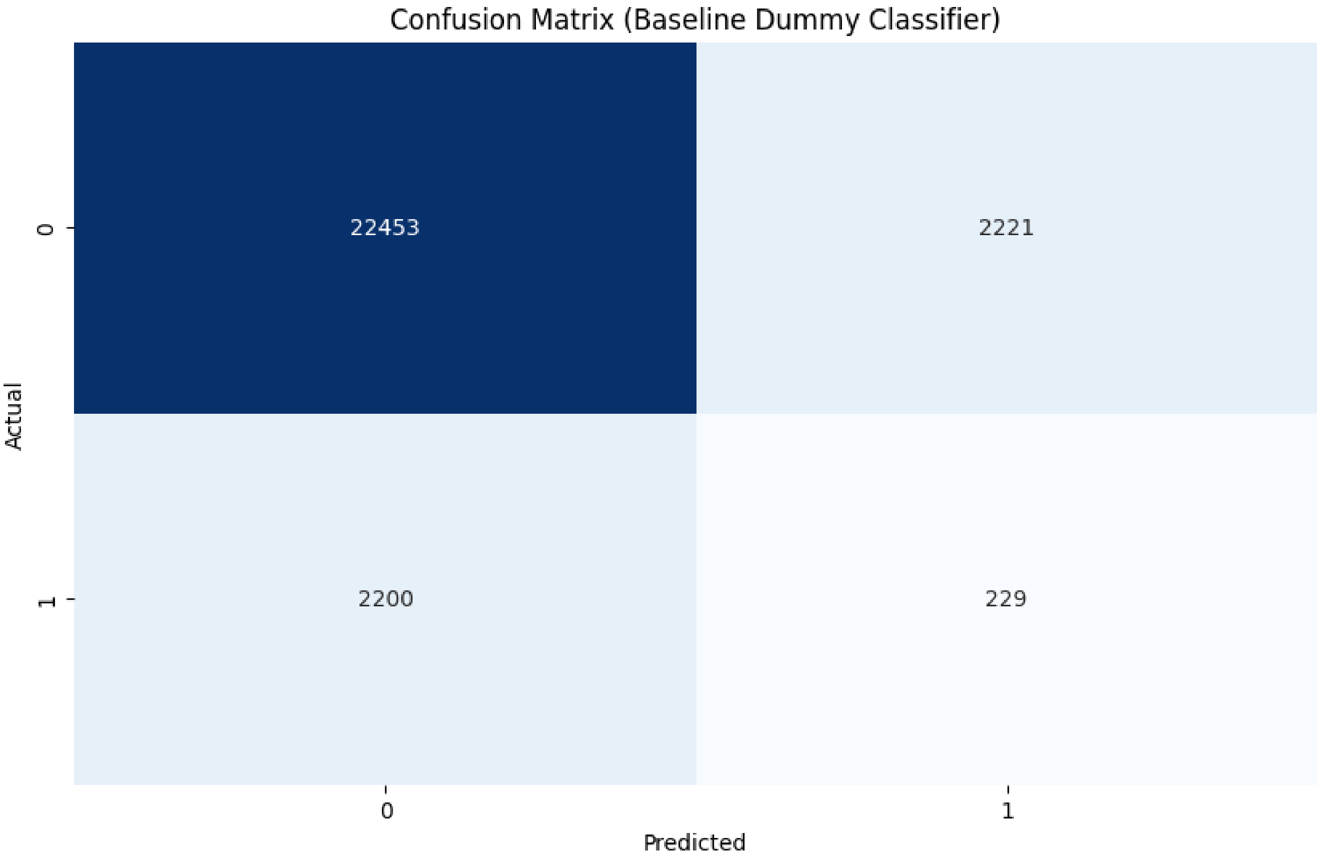


Figure 4: Confusion Matrix for the Baseline Model: The Baseline Model, unsurprisingly, classifies the majority class well, but fails to capture the minority class, resulting in a large number of false positives and false negatives. This reflects the metrics shown in Figure 3.

As stated in our project constraints, fraudulent transactions detected must be accurate at 90% accuracy or more. The baseline model's accuracy stands at 84%, and its accuracy at identifying fraudulent transactions is 9%. Therefore the baseline classifier does not meet our threshold and it is not a good model to deploy. In order to enhance accuracy, the team will be running multiple algorithms and generating a stacked ensemble. At this stage, we strongly advise GT Bank's leadership to increase their cloud capacity or grant us access to more hardware GPUs in order to train our models with a larger sample, or even the full dataset, and therefore capture more observations in the ML training process which could yield higher accuracy.

4 Model Selection

Obtaining a baseline model provided us with the necessary insights for model selection. Following the preliminary classification, we have decided to evaluate the following models: Logistic Regression, k-Nearest Neighbors, Decision Trees, Support Vector Machines, Gaussian Naive Bayes, Random Forests, Bagging, Gradient Boosting, and Extreme Gradient Boosting. The models were evaluated based on mean and standard deviation of ROC-AUC Scores from 5x3 Cross Validation.

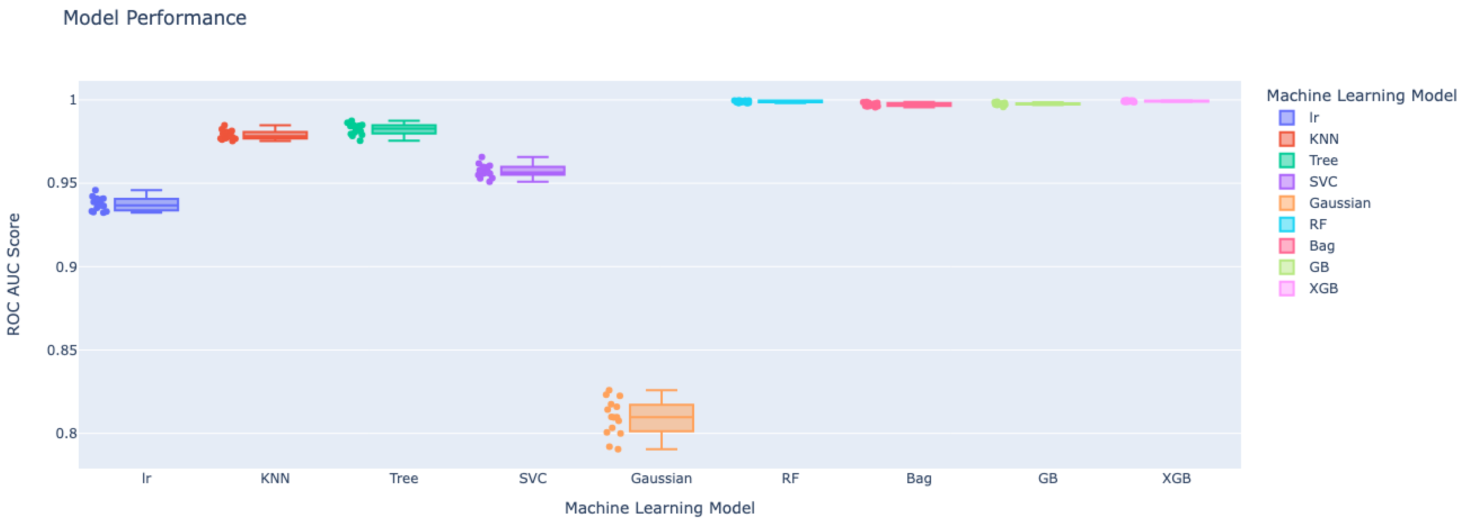


Figure 5: Machine Learning Model Results: The top 5 models are Extreme Gradient Boosting, Random Forests, Gradient Boosting, Bagging, and Decision Trees.

In an effort to improve accuracy and reduce overfitting, we advise GT Bank to also develop a stacked classifier, which would also make the model more robust to outliers and noisy data. Based on the results from Figure 5, the five models that will be considered as candidate models for level 0 in the stacked model analysis are Extreme Gradient Boosting, Random Forests, Gradient Boosting, Bagging, and Decision Trees. We will use Decision Trees as the level 1 combiner (or metamodel) to aggregate the results of the level 0 models. In addition to the selected models, we have also opted for cross-validation of the dataset. The models will then be evaluated based on mean and standard deviation of ROC-AUC Scores from 3x5 Cross Validation.

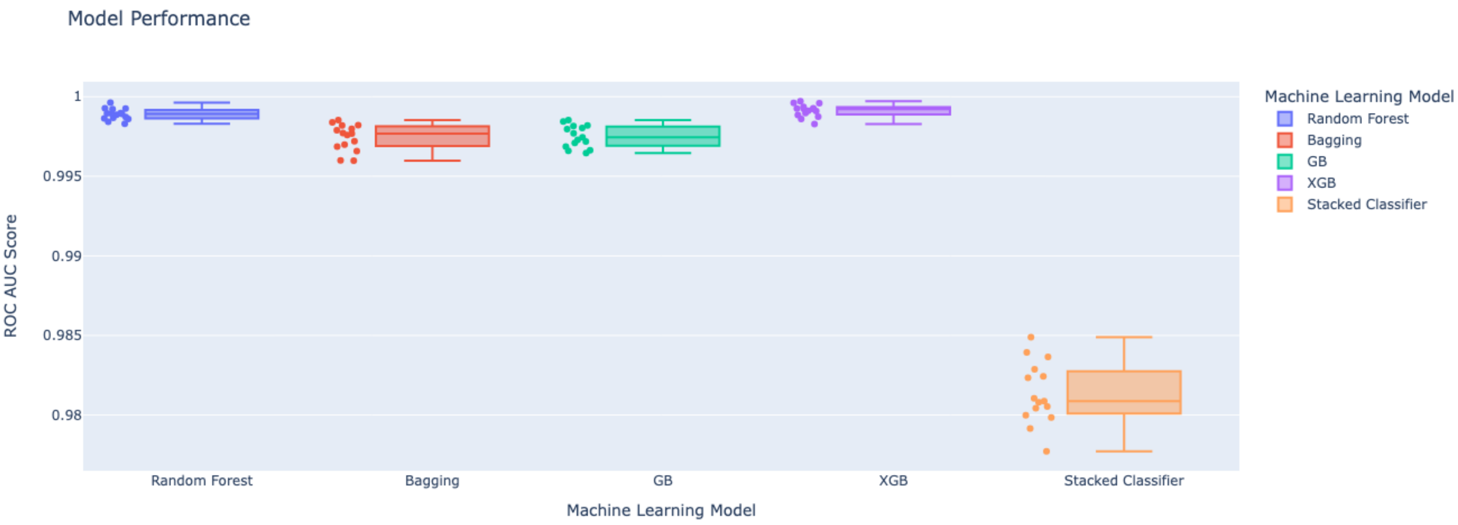
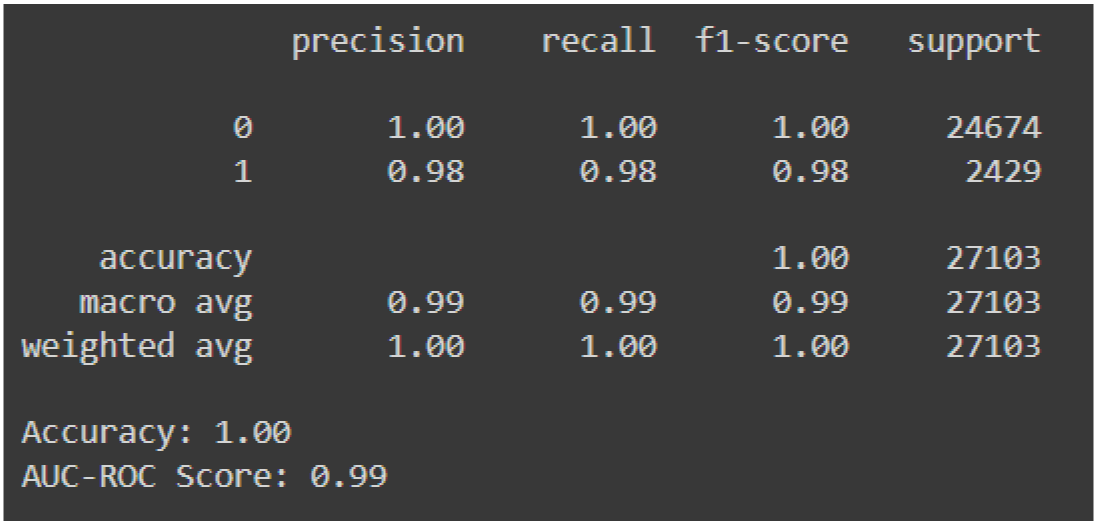


Figure 6: Candidate Model and Stacked Model Performance: Extreme Gradient Boosting and Random Forests performed the best, and the Stacked Model performed the worst.

Based on the results from Figure 6, we can see that the Stacked Model performed worse than the other candidate models. However, we will address this via hyperparameter tuning, and we will focus on credit card fraud classification by using the Stacked Model for the remainder of the analysis.

4.1 Model Performance Evaluation



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     24674
           1       0.98      0.98      0.98      2429

    accuracy                           1.00     27103
   macro avg       0.99      0.99      0.99     27103
weighted avg       1.00      1.00      1.00     27103

Accuracy: 1.00
AUC-ROC Score: 0.99
```

Figure 7: Model Metrics for the Stacked Classifier: Despite the imbalanced dataset, the Stacked Classifier shows good performance.

Based on Figure 7, the Stacked Classifier, despite the imbalanced dataset, shows great overall accuracy. We can also see good performance in the precision, recall, and f1-score for both the majority and minority class. Additionally, the AUC-ROC score, which would be a better metric than accuracy in interpreting the performance of a model on a heavily imbalanced dataset, is 0.99, which suggests that the model is able to discriminate between the two classes very well.

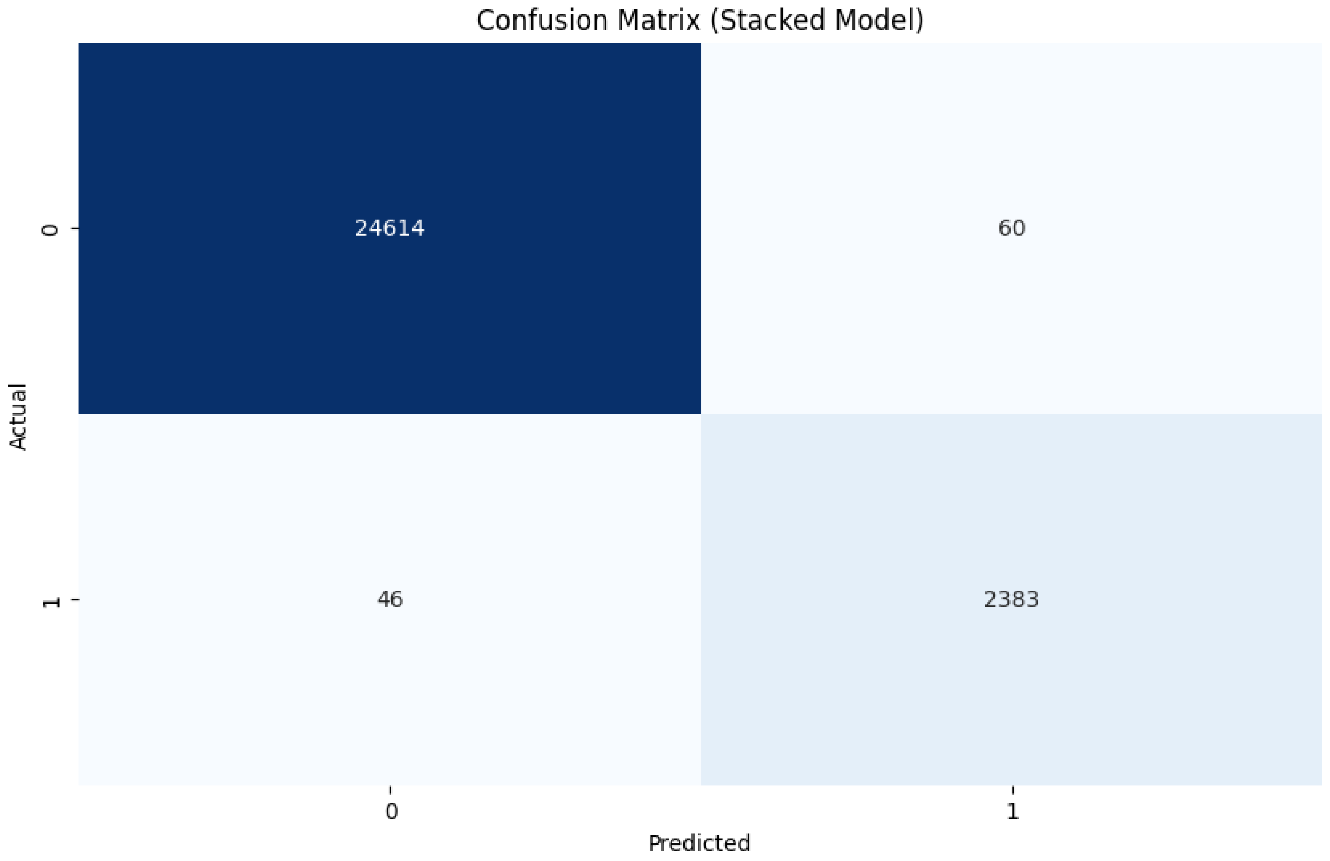


Figure 8: Confusion Matrix for the Stacked Model. The Stacked Model classifies both the majority and minority class well. There are still some false positives and false negatives, but the Stacked Model shows a great improvement compared to the baseline Dummy Classifier. This reflects the metrics shown in Figure 7.

At this stage, we are happy to inform leadership that the development of a Machine Learning model that can predict fraud with high accuracy is going very well. We have achieved success in exceeding our 90% target accuracy. However, we would also like to advise that the model could be further solidified with hyperparameter tuning, which will be our next step prior to deployment.

5 Initial Deployment

In order to proceed with displaying the most appropriate output for our application, the team has run a feature importance analysis. Figure 9 shows the average feature importance by feature. Since `oldBalanceOrig` and `newBalanceOrig` have the highest average feature importance, we'll plot these features against the dependent variable `isFraud`.

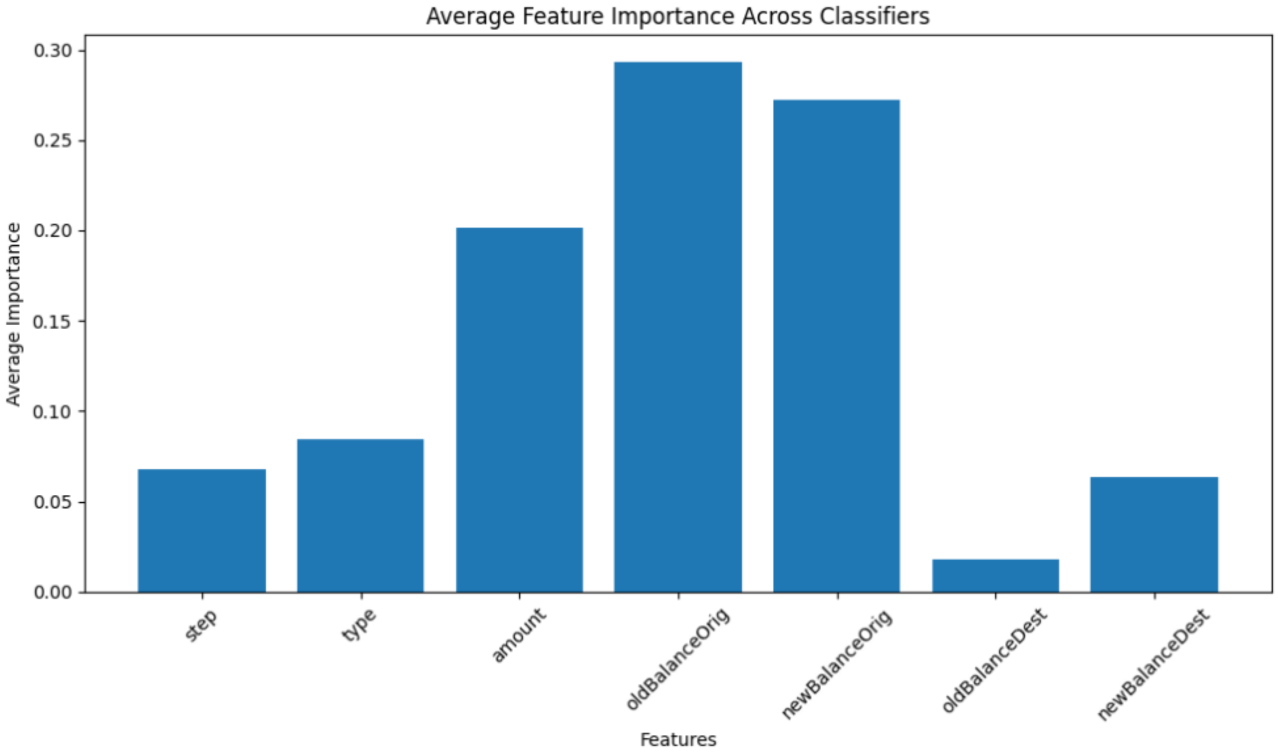
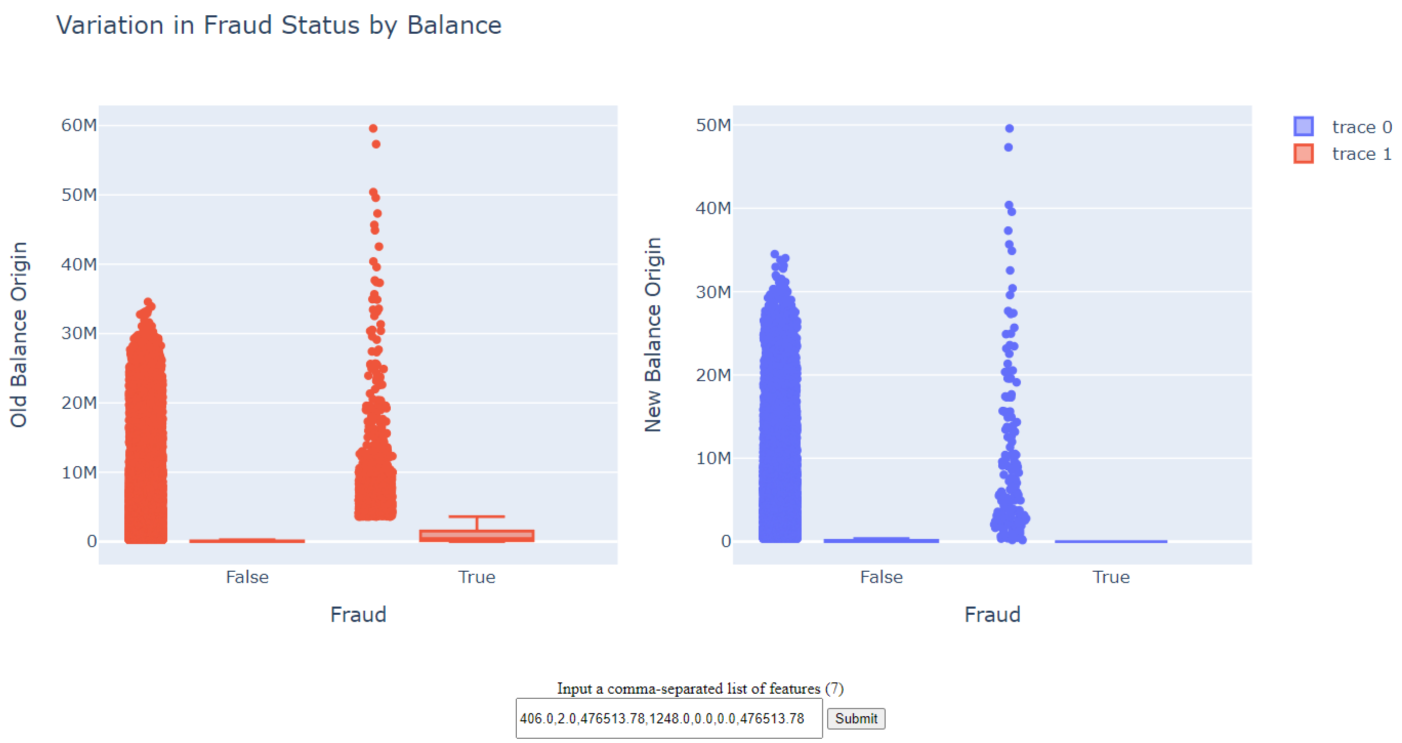


Figure 9: Feature Importance Graph. Our 2 most important features are oldBalanceOrig and newBalanceOrig.

We have selected Heroku as a final deployment site. The app will display two boxplots. The first boxplot will display the dependent variable against the feature `oldBalanceOrig`, while the second boxplot will display the dependent variable against the feature `newBalanceOrig`.

5.1 Screen 1

Figure 10 shows the initial plots of the dependent variable. Users can provide input to the text box below the image. This input is processed and then used to predict whether the given transaction is fraudulent or not.



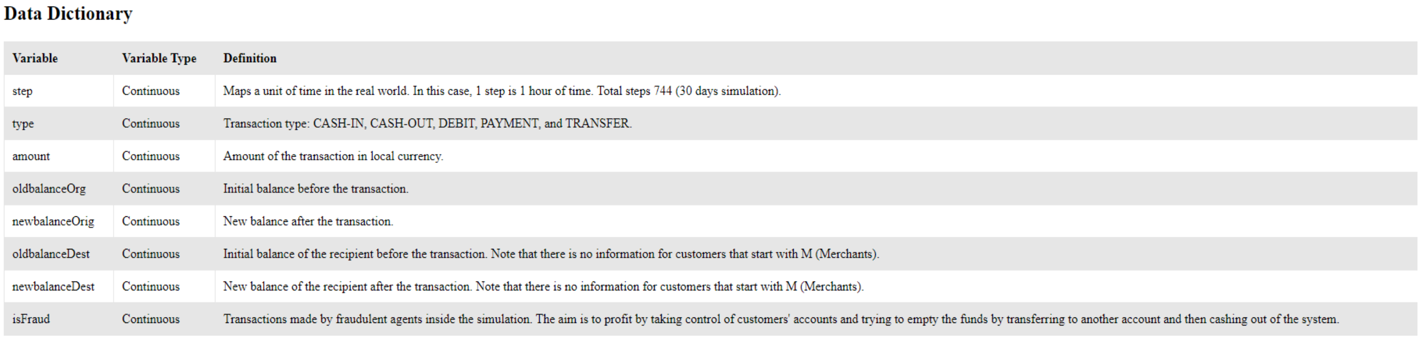
Input a comma-separated list of features (7)

406.0,2.0,476513.78,1248.0,0.0,0.0,476513.78

Figure 10: Screenshot of the unpredicted dataset

Figure 11 shows the data dictionary provided to the user on the app in order. The dictionary makes it easier for the user to understand the types of input that need to be entered into the box in order to generate the prediction.

Data Dictionary:



Data Dictionary

Variable	Variable Type	Definition
step	Continuous	Maps a unit of time in the real world. In this case, 1 step is 1 hour of time. Total steps 744 (30 days simulation).
type	Continuous	Transaction type: CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER.
amount	Continuous	Amount of the transaction in local currency.
oldbalanceOrg	Continuous	Initial balance before the transaction.
newbalanceOrig	Continuous	New balance after the transaction.
oldbalanceDest	Continuous	Initial balance of the recipient before the transaction. Note that there is no information for customers that start with M (Merchants).
newbalanceDest	Continuous	New balance of the recipient after the transaction. Note that there is no information for customers that start with M (Merchants).
isFraud	Continuous	Transactions made by fraudulent agents inside the simulation. The aim is to profit by taking control of customers' accounts and trying to empty the funds by transferring to another account and then cashing out of the system.

Figure 11: Data Dictionary. This provides a list of variables that feed the model, along with their types and definitions.

5.2 Screen 2

Figure 12 shows the updated image with the predicted price in green and orange on the plots. The feature vector for this prediction was:

406.0,2.0,476513.78,1248.0,0.0,0.0,476513.78

Other Options are:

1. 655,0,45758.71,45758.71,0.0,0.0,0.0,1,0
2. 655,1,45758.71,45758.71,0.0,0.0,45758.71,1,0
3. 656,1,190606.46,31936.0,222542.46,582386.77,391780.31,0,0
4. 656,2,2159.33,20094.0,17934.67,0.0,0.0,0,0
5. 656,0,99371.31,18404.0,0.0,2434200.04,2533571.35,0,0

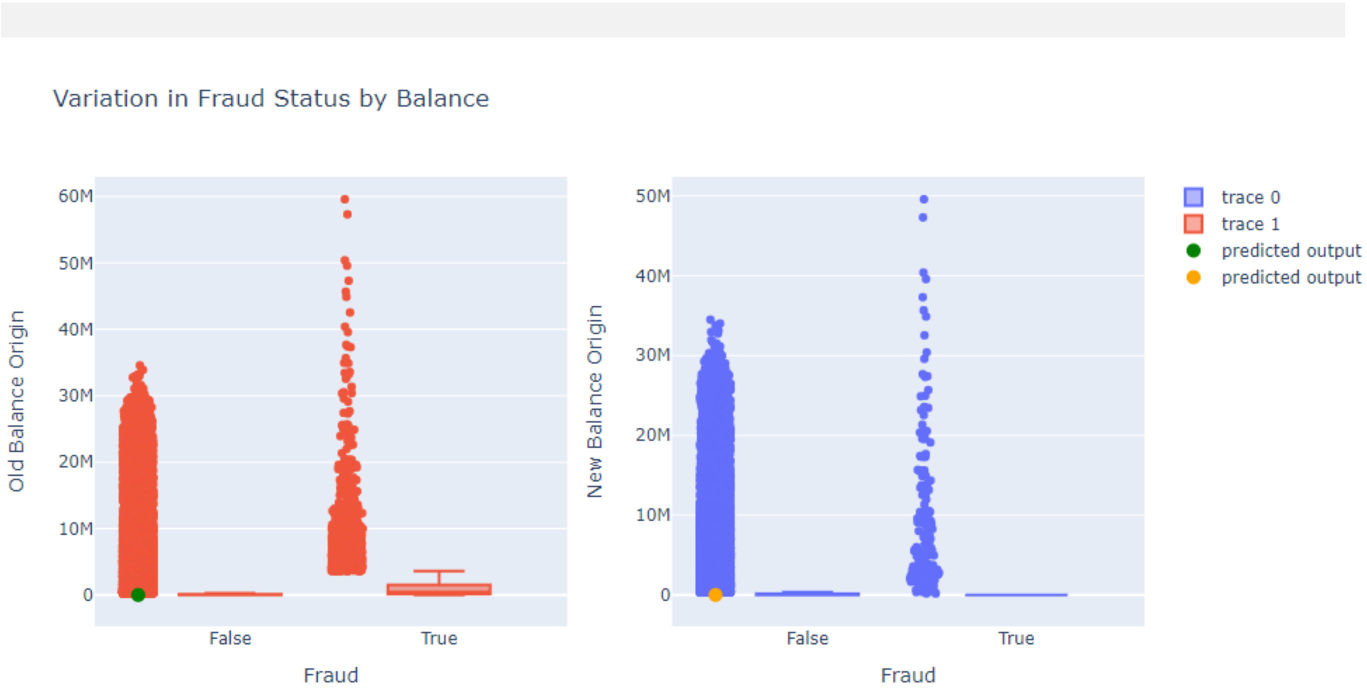


Figure 12: Predicted Vector. Based on the given vector, the model's prediction deems the transaction not fraudulent

5.3 Heroku Application

5.3.1 The Heroku Platform

5.3.2 Application path

5.3.3 Deployment, debugging, and updates

6 Conclusion

In conclusion, the successful development and deployment of our machine learning model to predict fraud in credit card transactions represent a significant milestone in addressing the challenges posed by fraudulent activities in the financial industry. The critical role that accurate fraud detection plays in safeguarding not only the interests of financial institutions like GT Bank but also the financial well-being and peace of mind of their customers cannot be understated. As we wrap up this project, we reflect on the importance of our endeavor and its implications.

As stated in our project introduction, the impacts of fraudulent transactions can be severe and far-reaching, affecting not only the financial institutions themselves but also individuals, businesses, and the broader economy. We recognize that GT Bank is committed to delivering a secure and satisfying banking experience to its customers. Our machine learning model has provided a robust solution to enhance the security of credit card transactions and mitigate the potential consequences of fraud. One of the most gratifying aspects of this project is our ability to achieve a high level of accuracy in our fraud detection model, exceeding our target goal of 90%. This achievement is a testament to the expertise and dedication of our team and the commitment of GT Bank in embracing innovative solutions. With the model successfully deployed on the Heroku platform, GT Bank can now proactively identify fraudulent transactions in real-time, providing timely alerts and mitigating the financial losses that can result from such activities.

However, as we move forward, we strongly recommend that GT Bank consider increasing its cloud capacity or providing additional hardware GPUs to further enhance the model's capabilities. By expanding our sample size or even training on the entire dataset, we can capture more observations, refine our model, and potentially achieve even better, more well-comprehensive results. With this investment, we can ensure that the model continues to adapt and improve its fraud detection capabilities over time.

In summary, this project has been a significant step forward in enhancing the security and customer experience of GT Bank in the face of fraudulent credit card transactions. We are confident that the successful deployment of our machine learning model will yield tangible benefits for the bank and its valued customers. By addressing the challenge of fraud in credit card transactions, we are not only safeguarding financial interests but also contributing to the broader goal of ensuring that the banking experience remains secure, seamless, and customer-centric.

Thank you for the opportunity to be a part of this important project, and we look forward to continuing to work closely with GT Bank to explore further improvements and innovations in the realm of financial fraud detection and prevention.