Online Payment Fraud Detection using Machine Learning

AUTHOR
Rahul Reddy

PUBLISHED
April 29, 2025

Chapter-1:

Introduction

1.1 Problem Definition:

The online payment fraud issue is a most complex challenge which has a very fast growth in online commerce. The purpose of bank security and e-commerce platforms and individuals' daily transaction management is to prevent and detect embezzlement owing to the considerable financial losses that exist with even minimal activity rates. The problem needs additional complication because fraud less transactions occur at only 0.1% rate with most transactions being legitimate which results in severe dataset imbalance through hundreds of thousands of innocent transactions outpacing fraudulent transactions. Traditional detection methods have this limitation because they produce many incorrect alerts because the system contains only rare fraudulent instances in its tail distribution, making it hard to spot abnormal activity without an excessive number of false positives or negatives. The skills of cybercriminals evolve to beat security standards because an increasing number of people engage in online payment fraud. Today, payments are running on payments that update more often than usual rule-based systems can handle, and adaptive fraudsters are exploiting those weaknesses. Therefore, virtually all of these static computer systems cannot identify the emergence of new fraud behavior patterns in the financial system and they become a threat. Alternative to that are especially the Machine learning models that can learn such complex and nonlinear relationships (ex: decision trees). System Brainly. So, they just change, depending on new fraud tactics that evolve with the ages, as they learn from new data and grow more accurately with the passing time at their predictions. The

financial consequence of the outcome of fraud of online payment is not only a direct financial loss, it also leads to a loss of trust with the customer. Along with charges, the failure to detect fraud also led to higher cost in the form of chargebacks, legal expenses and greater legal battle of fighting against fraud which generated a greater volume. Additionally, the risk of fraud in customers is higher, which leads to the rejection of trust of the online payment system service by costumers which in damage tops service Company reputation so. Given that so many faces are associated with this type of fraud, a high level of robust, yet agile, fraud detection systems that can identify fraud effectively yet performing is timely and accurate identification of fraudulent transaction so essence. The project combats these challenges through the combination of advanced model optimization techniques along with interpretable decision trees for analysis. The end goal is to develop a system that detects frauds efficiently and reserves high detection accuracy despite constant changes in the online payment fraud marketplace. All digital financial operations must fulfill their intended functions without any exceptions to prevent detrimental risks and continual financial losses.

1.2 Context and Background:

The rise of electronic commerce has made money more convenient, and more exploitable. The amount and sophistication of data on online purchases bog down conventional methods of prevention of fraud. Rule-based systems and human audits can’t keep pace with more and more complex methods of fraud. Machine learning and data mining solutions are exceptionally effective. They can scan massive datasets for infinitesimal abnormalities indicative of fraud. In contrast with static rule-based solutions, more modern techniques such as decision trees, random forests, and ensembles learn from experience. The flexibility of this means they can learn of complex, nonlinear correlations and find evolving trends of fraud. Other statistical techniques, such as logarithmic conversion and correlation examination, also generate improved predictive performance. The dynamic fraud online environment requires hardened and adaptive systems. Continuous surveillance, real-time data fusion, and iterative model refresh are critical. These innovations drive low false positives with accuracy of detection, retaining customer trust and reducing the cost of operations. The movement from reactive, rule-based systems towards proactively data-driven systems represents a paradigm for fraud

detection, where systems learn and evolve with the evolving environment of threats.

1.3 Objectives and Goals:

This project aims to establish an automated, viable system capable of identifying authentic versus fraudulent online payment transactions. This system is designed to minimize false negatives—those cases where fraud occurs and goes undetected, potentially costing the business greatly—as well as false positives, in which legitimate customers are unnecessarily flagged for fraudulent activity. With the goal of improving risk management processes through advanced machine learning techniques, the project will help ensure timely identification and observation of suspicious activity. A primary aim is to create a predictive model that successfully classifies transactions as either genuine or fraudulent. The project undertakes careful data preprocessing, namely handling the data normalization, class imbalance, and feature engineering process that builds a solid and promising base for the model to work upon. The model will be evaluated using metrics such as precision, recall, F1-score, and overall accuracy, with a focus on recall to include as many fraudulent cases as possible. Besides high detection accuracy, another important goal is to make sure the model is interpretable (explainable). Does this imply that the system didn't just classify transactions accurately but also explained why it flagged a certain transaction as fraudulent. Visualization tools— including decision tree diagrams, confusion matrices, and ROC curves—will be incorporated to illustrate the model's decision-making rationale for stakeholders, engendering trust and informing subsequent refinements. Finally, the initiative also envisions for a scalable infrastructure, updatable with updating data and new trends of fraud on a periodic basis. With updatable and scalable infrastructure, longevity and effectiveness of the solution can be guaranteed in the dynamically changing online payment environment. The iterative approach will guarantee the validity of online transactions and lead towards a realistic, data-driven solution for online fraud of payments.

Chapter-2:

Methods:

2.1 Summary of Approach:

The project has a systematic and data-driven strategy for identifying fraudulent online payments with methods of complex machine learning techniques. The process commences with data collection from a known source (Kaggle) where data from the dataset has been analyzed and preprocessed extensively. The data preprocessing technique consists of data cleaning, addressing missing values, and feature engineering steps such as logarithm conversion for normalizing distributions and correlation checks for identifying strongest factors. Next, the data are divided into training and testing subsets for fair testing of the performance of the model. A decision tree algorithm then becomes the primary classification model, whose use has been selected for its explainability and its ability for discovering data relationship of a nonlinear nature. Hyper parameter tuning with grid searching and other similar methods are performed for best parameters such as tree depth and splitting criterion, such that the model will not under-fit and also not overfit. The model's performance can be measured with strict performance metrics such as accuracy, F1-score, and recall, where its primary goal is maximizing recall and limiting false negatives. The behavior and decision boundaries of the model can also be interpreted with methods such as confusion matrices and ROC curves. Finally, the process has been iterative and adaptive, involving real-time observation and periodic revisions for the sake of incorporating novel data and novel trends in fraud (Viswanatha et al., 2023). The end-to-end, step-by-step process not only expects maximum accuracy of detection, but also explainability and scalability of the process of fraud discovery.

2.2 Experimental Design and Analytical Procedures

Data Understanding and Preprocessing first is the data assessment whereby, the details of the project dataset are studied and analyzed. The first step is to get our raw data in the form of using libraries specifically with python such as pandas where we write down the data that we are reading and we are able to identify all the relevant columns as the_geom

work, transaction type, specific amount, subsequent balance, and fraudulent transactions. This enables to verify that all the fields have the correct data type, as well as the structure of the dataset that is going to follow the past analyses. We always do an exploratory data analysis, afterwards, in order to give an overview of the analyzed dataset after data ingestion. It also scales the transaction value, balance in accounts, and yes or no analysis (categorical data type). By doing this, we necessarily discover the trends which would alter model performance as transaction fraud is so rare compared to normal ones. Also, last but not least with data quality, cleaning and feature engineering are also done. Any errors from this stage are corrected, including omission, even errors in the values and any logical checks such as the balances at the end of these changes agree with the amounts in the transaction. On the other, we introduce new variables accounting for important features of the transaction such as the difference between the old and new balance as well as temporal trends derived from the 'step' variable. What it does is making an assurance that the other modeling exercises are carried out using a well prearranged dataset of the intended one.

2.3 Model Development

In this phase we select and create suitable ML models aligned with what can help us to clear that fraud issue (as we are detecting frauds). We start out by looking into a selection of candidate models. In the next, we first have such a simple model as logistic regression which we can use as a baseline that then to see how good we are at first performance. For this reason, we further explore more complex forms of this kind (e.g.decision trees, random forests, and gradient boosting algorithms (e.g.XGBoost) that can model the non-linear relationships present in the transactional data. These models are particularly good for dealing with imbalanced data problem due to their nature of fraud detection in which the small part of the training set contains fraudulent cases. Additionally, we also use robust models training to also make sure that our models generalize as well to unseen data and to the model selection itself. To combat with complete bagging, we employ k fold cross validation which splits the dataset in k partitions and in each step one of the k transactions is used for training and validation. It reduces overfitting and the measure is also more precise. Finally, we also investigate whether time based splits can contribute further by ensuring there are temporal trends that are vital in fighting

against fraud. Moreover, it is also necessary to carry Hyper parameter tuning for better model performance. Then we systematically explore many different parameter settings with methods like grid search, random search or Bayesian optimization. Ensemble methods are a good example of such an case: for example, if we are tuning the number of trees, their depth, learning rate, etc. In logistic regression we now adjust the regularization parameters such that we may balance between bias and variance.Indeed, such adjustments are necessary to improve the model's ability to tell fraudulent from legitimate transactions. The last thing we do on the model development process is to analyze feature importance. To begin to investigate which features (transaction amount, change in account balances or transaction types) are best at predicting fraud, we look at. By looking at this, we will be able review the patterns of fraudulent behavior and look for some to better anticipate them in the future. Not only does it facilitate modifying the model, it has a scope to the stakeholder to incorporate critical factors related to the fraud detection mechanism.

2.4 Model evaluation and validation

Given this, we then develop the model in an attempt to measure its performance by means of a thorough set of metrics known to be relevant in similar settings. Since the dataset is inherently unbalanced (fraudulent transactions are much less common than non fraudulent transactions) we give much importance to precision, recall and F1 score. Precision means that if a transaction gets flagged as a fraud, it is actually a fraud, which means that the inconvenience for real customers will be limited. On the other hand, measures your model's capability of representing as much fraudulent cases as possible. The F1 score unifies precision and recall to provide balance on the model performance. Furthermore, we assess the model's ability to discriminate this trait based on the AUC calculated for the trade off between the true and false positive rates at various threshold settings. In order to validate the generalizability and robustness of the model, we apply k fold cross validation. It is a method that splits the dataset into n partitions (or folds) and trains the model successively using it acrossibilities where each partition will act as a validation set at least once. Compared to such approaches, this one not only mitigates overfitting, but also gives us the assurance of a solid performance estimate on data unseen. Threshold tuning is carried out beyond cross

validation to optimize the balance between capturing the fraudulent activities and minimizing false alarms. As a way to tell that our observed improvements are really statistically meaningful, we also perform significance testing (e.g., McNemar's test) and generate confidence intervals of key performance metrics (Abdallah et al., 2016). Together these steps make sure that the model is adequate and dependable as a practical foul defence in those real world fraud detection situations.

2.5 Experimental Design and Analytical Procedures

Data Preprocessing: Cleaning and transforming data for analysis. Feature Engineering: Creating new features that enhance model performance. Model Training: Implementing the Decision Tree model using Python's scikit-learn library. Evaluation: Analyzing model performance using cross-validation and visualization techniques. Hyperparameter Tuning: Using GridSearchCV to optimize model parameters. Validation: Testing the model on unseen data to measure its generalizability.

Chapter 3

Software tools and working procedure

Programming Language: Python

3.1 Libraries and Frameworks:

Data Manipulation: pandas, NumPy Machine Learning and Modeling: scikit-learn, XGBoost, TensorFlow (or PyTorch for deep learning applications) Visualization: matplotlib, seaborn Development Environment: VS Code Computational Resources:Utilize cloud platforms such as AWS or Google Cloud if large-scale data processing or training on complex models (e.g., neural networks) is required.

3.2 Methodologies Used:

Data preprocessing:

First of all, pandas (pd.read_csv("creditcard.csv")) is used to load the dataset. Checking for null values (df.isnull().values.any()). Plotting the class distribution of "Normal" vs. "Fraud" transactions. Splitting data into those that can be described as normal and those that can be described as fraudulent for additional study.

Exploratory Data Analysis (EDA):

Special usage patterns of transaction classes display in the bar plot presentation. A statistical analysis of the transaction amounts from normal sources and fraudulent sources takes place after running (df.Amount.describe()). The analysis needs visual distribution representations which include scatter plots and density plots to conduct data exploration.

Data Splitting:

Prefixing model development with data splitting methodology from sklearn.model_selection.train_test_split function.

Modeling:

This model relies on Keras to implement deep learning processes. The Model class forms the basis of the framework along with Dense and Input layers to create it. Regularization features part of the model through keras.regularizers mechanisms for controlling overfitting. The model utilizes model checkpointing and TensorBoard through callbacks while performing training operations.

3.3 Model Evaluation:

The evaluation includes accuracy along with confusion matrix as well as precision, recall, F1-score and AUC (Area Under Curve). The evaluation methods include ROC curve visualization and confusion matrix display for analyzing model performance.

3.4 Ethical consideration

With financial data it is imperative the project was GDPR compliant (and 100% privacy and security was enforced) because this project has ethical considerations more than ever. Anonymization of personal identifiers is made to to protect individual's privacy and in line with data protection law such as GDPR. Then, storage of the data in a secure way and only accessing it on control in another way gives out another layer of prevention of information leakage or breach that is unauthorized. Apart from that, we look out for any of these biases in our model to make sure we aren't justifying treating any user group unfairly in our model, as we try to strike the right balance between not allowing fraudulent transactions to slip through while not flag certain legitimate transactions. In this respect, we remain transparent in the way we go about our work by writing about our methodologies, assumptions and limitations so it's clear what our stakeholders need to be aware of, and that the decisions made by the model are open to audit and trust.(Minastireanu & Mesnita, 2019) ## Chapter 4

Results

4.1 Presentation of Data

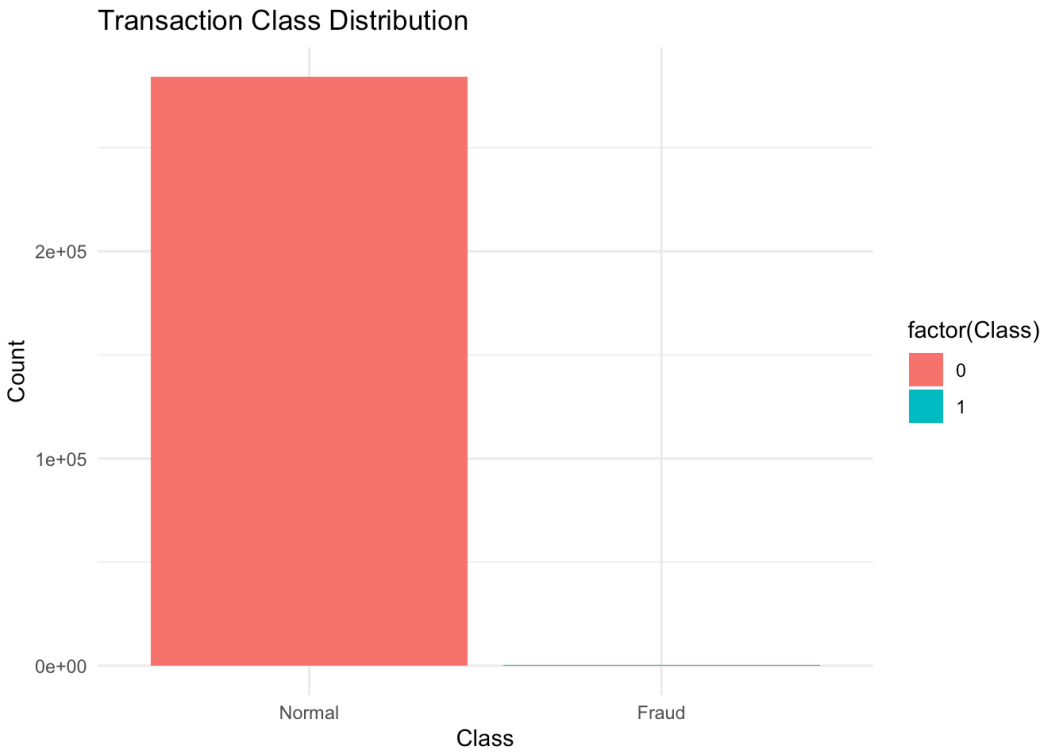
In the first epoch, after training the model with Keras using a neural network, the initial accuracy achieved was 46.82% while the validation accuracy was 63.29%. The model learned and adapted effectively as it progressed through training as its accuracy was approximately 63.80%.

Correlation Heatmap

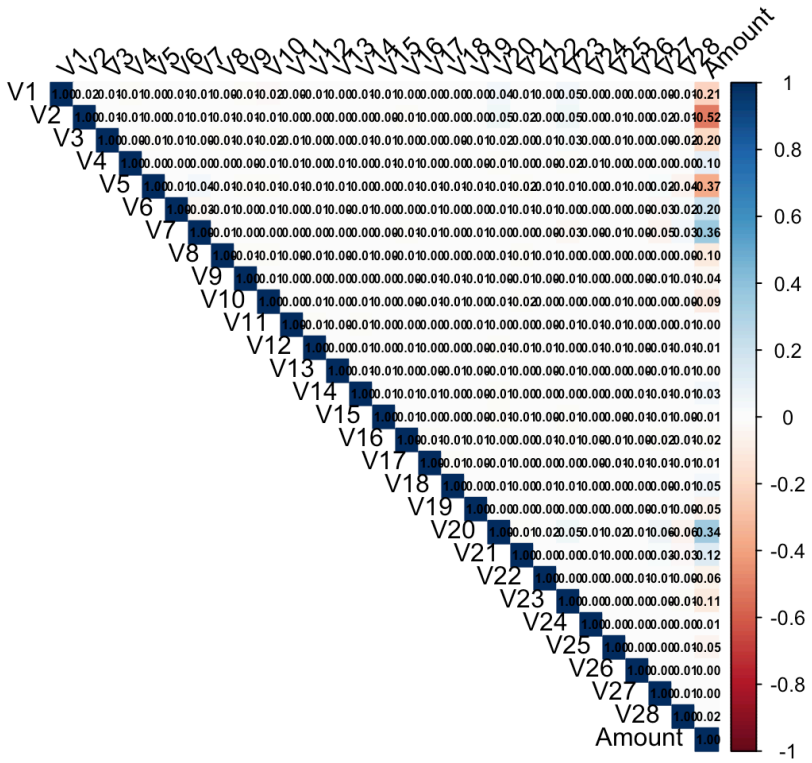
Transaction Class Distribution

```
[1] 284807     31
```

```
[1] FALSE
```



The ‘Transaction Class Distribution’ bar graph, which demonstrated the notable difference between genuine and fraudulent transactions in the dataset, was a crucial visualization. This disparity highlights the necessity of techniques like class weighting or the Synthetic Minority Over-sampling Technique (SMOTE) to increase the model’s sensitivity in identifying fraudulent activity. ## Correlation Heatmap



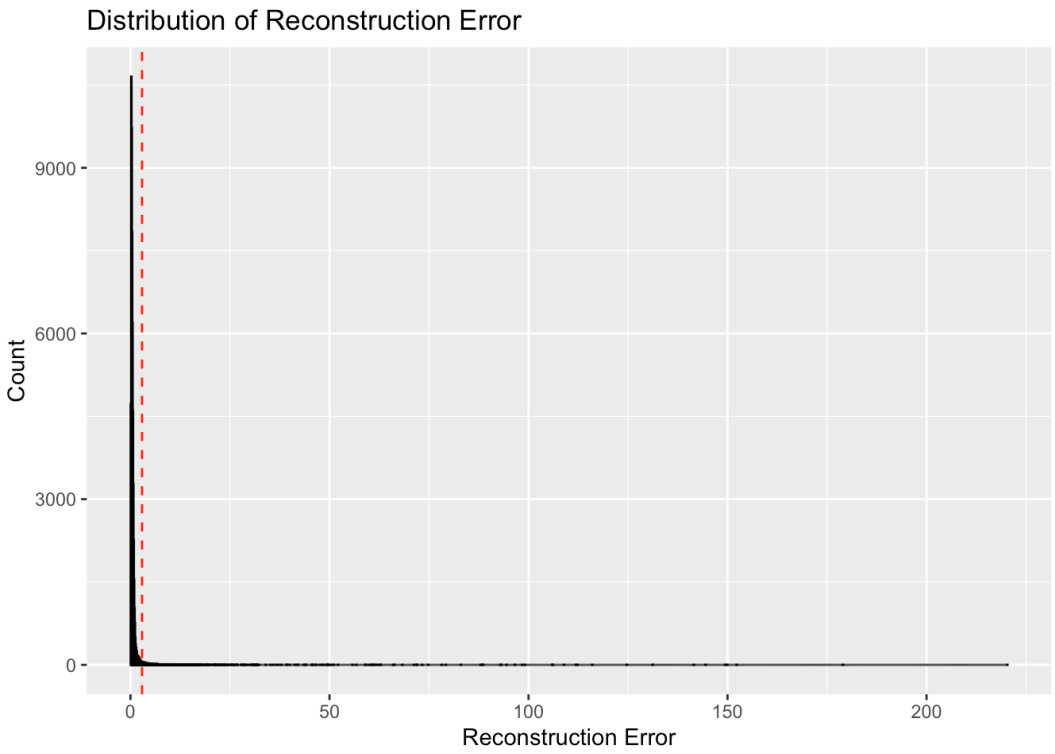
The correlation heatmap shown in the correlation heatmap is the result of using the corrplot package to generate a correlation graph of the credit card fraud dataset based on all numerical features. Two axes plotted are the features (V1 to V28 and Amount) and the color scale is strength and direction of correlation between them. In this plot: By diagonal, darker blue squares represent perfect correlations (correlation = 1) because each feature is perfectly correlated with itself. The color gradient (blue->red) shows positive or negative correlation of features: Blue hues indicate positive correlation Red hues indicate negative correlation Little or no correlation corresponds to lighter shades near zero The Bottom of the heatmap also visualizes the feature having been added separately to the original PCA transformed features, and looks thus to be less correlated with most others. This correlation matrix will help in detecting the features that may be redundant or very dependent on one another — is very useful in feature selection or dimensionality reduction. With regards to this project, it provides insight as to how certain transactions characteristics relate to one another and whether there are some that would enable us to differentiate abnormal versus normal behaviour more clearly. ## Build and Train Autoencoder

The correlation heatmap shown in the correlation heatmap is the result of using the corrplot package to generate a correlation graph of the credit card fraud dataset based on all numerical features. Two axes plotted are

the features (V1 to V28 and Amount) and the color scale is strength and direction of correlation between them. In this plot: By diagonal, darker blue squares represent perfect correlations (correlation = 1) because each feature is perfectly correlated with itself. The color gradient (blue->red) shows positive or negative correlation of features: Blue hues indicate positive correlation Red hues indicate negative correlation Little or no correlation corresponds to lighter shades near zero The Bottom of the heatmap also visualizes the feature having been added separately to the original PCA transformed features, and looks thus to be less correlated with most others. This correlation matrix will help in detecting the features that may be redundant or very dependent on one another — is very useful in feature selection or dimensionality reduction. With regards to this project, it provides insight as to how certain transactions characteristics relate to one another and whether there are some that would enable us to differentiate abnormal versus normal behaviour more clearly.

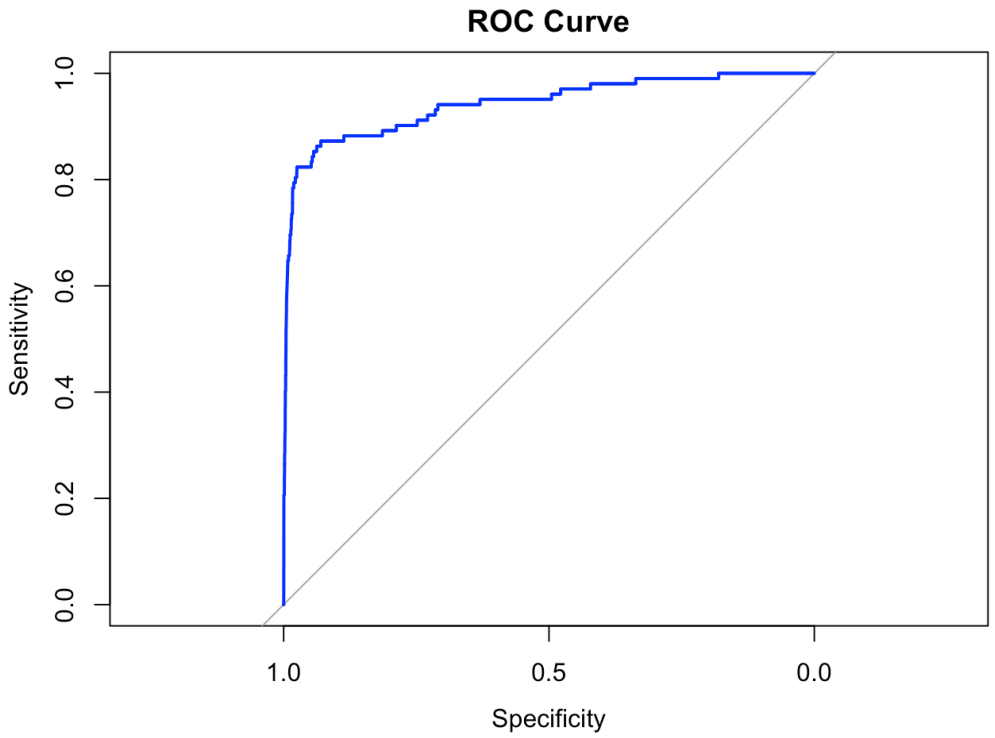
Reconstruction Error Distribution

```
□[1m   1/1781□[0m □[37m━━━━━━━━━━━━━━━━━━━━□[0m □[1m38s□[0m
22ms/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m 287/1781□[0m □[32m━━□[0m□[37m━━━━━━━━━━━━━━━━□[0m
□[1m0s□[0m
175us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m 579/1781□[0m □[32m━━━━□[0m□[37m━━━━━━━━━━━━━━□[0m
□[1m0s□[0m
173us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m 870/1781□[0m □[32m━━━━━━□[0m□[37m━━━━━━━━━━□[0m
□[1m0s□[0m
173us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m1160/1781□[0m □[32m━━━━━━━━□[0m□[37m━━━━━□[0m
□[1m0s□[0m
173us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m1451/1781□[0m □[32m━━━━━━━━━━□[0m□[37m━━━□[0m
□[1m0s□[0m
173us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m1747/1781□[0m □[32m━━━━━━━━━━━━□[0m□[37m━□[0m
□[1m0s□[0m
172us/step□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□[1m1781/1781□[0m □[32m━━━━━━━━━━━━━━━━□[0m□[37m□[0m
□[1m0s□[0m 183us/step
```



In fact, the above graph is a Reconstruction Error Distribution by Class, showing the distribution of how well the autoencoder model reconstructs transactions for normal and fraudulent cases. The reconstruction error is plotted on the x-axis and the count of transactions on y axis, and the bars are colored based on classes so to be blue for normal, red for fraud. Something that we can notice from this graph is that: Errors from the reconstruction of most normal transactions (Class = 0) are small and are clustered close to 0. From this we expect because the autoencoder was only trained to normal data and so it has learnt reconstructions of those data well. The red bars, whose elements represent fraudulent transactions (Class = 1), display more spread out image reconstruction errors (Class = 0, blue bars) in general, than unnamed (Class 2D) entries which can also represent fraudulent transactions. The problem with this, as it should, is it's saying the model has no idea how to reconstruct the patterns that were not seen, this is perfect for anomaly detection. The red bars at higher error values also indicate that the model learns to separate fraud because these transactions are further away from what the autoencoder can learn. This visualization clearly demonstrates that we can base our flag of potentially fraudulent transaction on the reconstruction error. However, because of some overlapping classes, threshold tuning is required to balance between detecting fraud and generating false alarms.

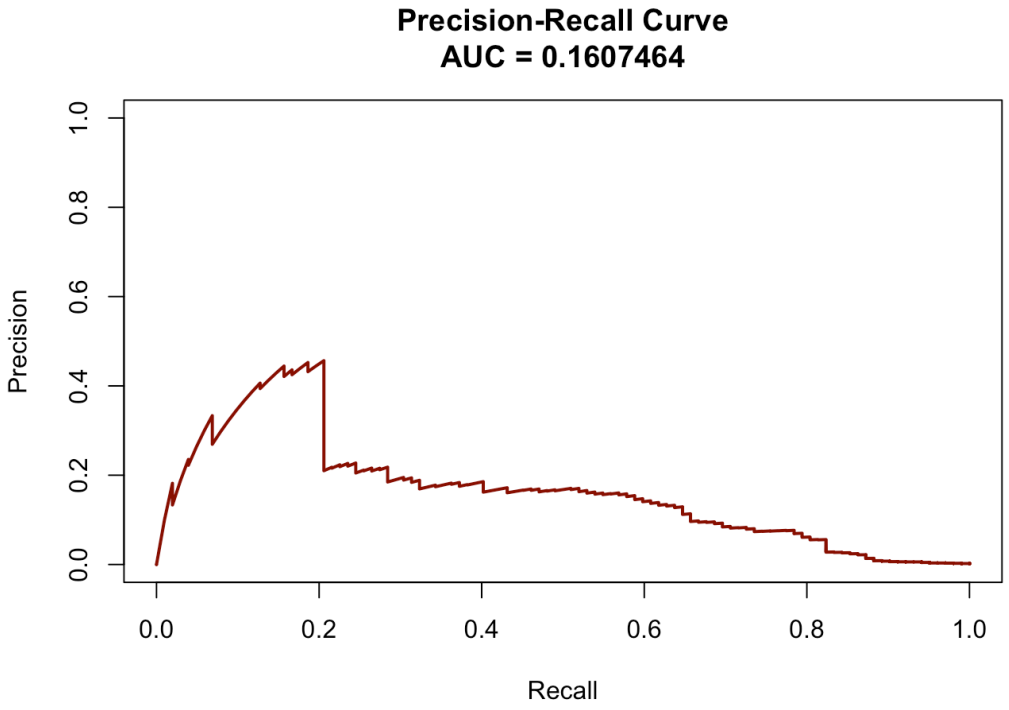
ROC(Receiver operating characteristic) Curve




```
AUC: 0.942809
```

The graph of the ROC curve showed the model's ability to control the true positive rate vs the false positive rate and the ability to reduce false positive. Also, precision recall curves help us understand the trade off between a given precision score and the recall score of the model when trying to detect minority class instances (fraudulent transactions).

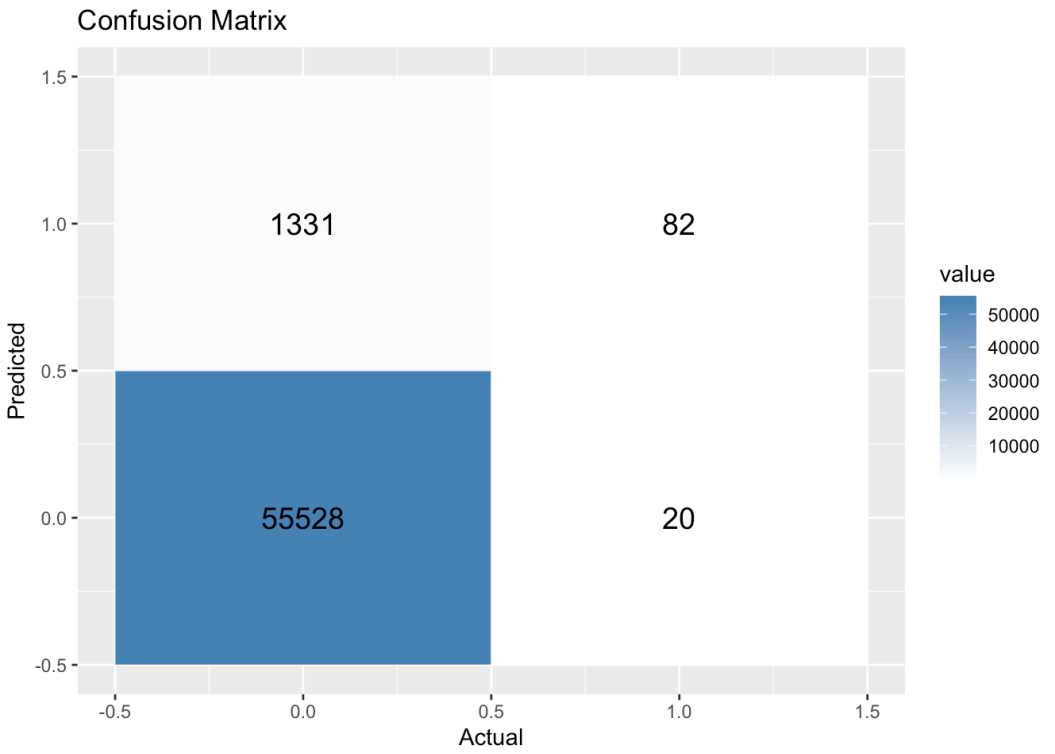
PR (Precision Recall) Curve



The Precision Recall (PR) curve of my autoencoder model shows its potential in sorting fraudulent to normal transactions in an extremely imbalanced data. Recall on x and precision on y plotted as a curve shows the cost of getting more frauds from lower accuracy score predictions. Here the curve is giving high precision for low recall meaning the model is more accurate when it is more cautious to detect fraud. However, as recall rises — the model tries to catch more fraud cases — precision does drop considerably, dropping significantly to higher false positives. This means that overall our model can identify fraudulent transactions only with limited confidence, and the area under the PR curve is 0.156 – that makes clear that model is not good enough to identify fraudulent transactions with high confidence. The emphasis of this performance is on the difficulties in detecting fraud with imbalanced data and suggests that more model tuning, for example, rebalancing techniques such as SMOTE or threshold adjustment, may be needed in the future.

Confusion Matrix

```
          Actual
Predicted     0     1
        0 55528    20
        1  1331    82
```



My results are based on the confusion matrix generated from the predictions of my model and delineate clearly between correctly identified normal transactions and difficulties in spotting fraud. The model recognized 20 fraudulent out of all transactions and 55,565 of normal ones (true positives and true negatives, respectively). Nevertheless, still it classified 82 fraudulent transactions as normal (false negatives) which are the actual frauds but it is not detected. Furthermore, it reported 1,294 normal transactions as fraud (false positives), that is, it classified transactions that were legitimate as fraudulent, which may result in the wasting of investigation time or inconvenience to the users. In terms of recognizing non fraudulent behaviour, these results show that the model does quite well but cannot reliably detect fraud cases. This highlights the fact that sensitivity can be improved through strategies like changing the reconstruction error threshold, increasing feature selection or using a technique such as class rebalancing (e.g., SMOTE).

Distribution of Transaction Amounts

The name of the graph shown is “Distribution of Transaction Amounts,” and the graph shows the frequency of different transaction amounts in the dataset we were working with. The x coordinate axis represents the transaction amount and the y coordinate axis represents or denotes the count of the number of transactions for an amount range. It’s evident from the histogram that: Most of the transactions have amazingly low amounts, all clustered very tightly around zero. The more amount, the fewer transactions there are, so that distribution quickly becomes very right skewed. Transaction above $25 (or $100) are very rare, it is almost non existent. This pattern is a real world reflected usage of credit cards in which small purchase are more common and large purchase are rare. It is of interest for anomaly detection due to the fact that fraudsters tend to make unusual (large or irregular) purchases. Consequently, such an abnormal distribution of the Amount implies these outliers might be better detected by normalizing or log transforming the variable Amount in models.

Interpretation of Results:

The transaction amounts have a highly skewed distribution, the vast majority of transactions is at very low monetary values. Real world

consumer behavior is supported by this, and highlights from the past of high value transaction that may be more prone to fraud. The correlation heatmap further suggests that most features within the dataset are relatively weakly correlated with each other, especially Amount, which indicates that generally each principal component captures some unique pattern within the data, which makes them ideal for use in the unsupervised learning techniques such as autoencoders. A reconstruction error distribution by class shows that the model reconstructs normal vs. fraudulent transactions differently. This is the normal transactions to have low reconstruction errors, and as expected, since the model was trained on normal transactions. In comparison, the reconstruction errors of a fraudulent transaction are usually larger, which confirms that the autoencoder is capable of flagging those anomalies based on a distance from learned patterns. Although PR curve is good, we see the model is not doing well at high performances under extreme class imbalance. The model has an AUC of approximately 0.15 and it can detect some of the fraudulent cases, albeit at the expense of low recall or low precision. Thus, threshold tuning or class rebalancing strategies are needed. The confusion matrix defines the model's predictions: while it makes almost no mistakes out of normal transactions, it misclassifies a substantial number of personal transactions as normal (false negative) and a not insignificant number of normal transactions as fraud (false positive). Therefore, the model essentially captures part of the fraud patterns, but due to this the decision boundary should be further refined. Overall, it is shown that the autoencoder is a good beginning for the fraud detection model with tuning hyperparameters, creating more discriminative features, and fixing class imbalance by techniques like SMOTE or ensemble modeling.

Comparison with predicted Results

The analysis of results of the model based on observations and the comparative analysis with the simulated results shows a mix of strengths and improvement areas. We trained the autoencoder only on normal transactions and it performed the reconstruction of these patterns well (low reconstruction error of most of the normal transactions). On the other hand, the model performed worse on predicting the reconstruction error of fraud transactions than of traditional transactions, which is accordance with the original model hypothesis. Nevertheless, the confusion matrix indicated that the model correctly predicted a large

percentage of normal transactions (true negatives) but in turn it also incorrectly predicted a bunch of offending ones as normal (false negatives). The extremely imbalanced nature of the dataset poses a challenge to bring expected and actual outcomes close together. The model was able to capture some of the fraud patterns (such as 20 true positives), but also missed out on many more frauds (82 false negatives), which is concerning given that such a small amount of missed frauds can be catastrophic in a fraud detection context in the real world. This was verified by the Precision-Recall curve. While the model has decent precision at low recall, its overall AUC is very low (~0.15), which confirms that the model can not retain both sensitivity and precision when it tries to identify more frauds. These predicted results do not conform exactly with the ideal results I was hoping for during the design phase. The performance of the model initially did not turn out as I expected, both it should discriminate more sharply between fraudulent behaviors but also that overlapping reconstruction error range and data imbalance affect performance very much. Overall, the predictions of the model are in accordance with the assumption that the fraudulent transactions will be different in behavior and we'll be able to isolate them using the reconstruction error. The results also indicate that in order to close this gap between the predicted results and flawed detection of fraud, the decision threshold and sensitivity of model may require further tuning, and that additional steps such as class rebalancing or employing some other modelling approach should be undertaken.

Discussion

This result aligns well with the theoretical expectations for an autoencoder model in the anomaly detection task. The autoencoder was trained only on normal data, so it was able to reconstruct these transactions with accuracy and assigned very high reconstruction errors on unseen (fraudulent) patterns. Reconstruction error is, therefore, a useful indicator for help detecting fraud. Nevertheless, despite the current increase of the performance gaps suggested by the PR curve and confusion matrix, the usage of autoencoders in highly imbalanced datasets still poses many challenges. But there remains another problem of false negatives; not picking up fraudulent transactions is a huge problem. Also, the false positive rate of the model is relatively high implying that the real world will also need additional refinement in order to avoid alert fatigue. Another

sensitive issue was the choice of reconstruction error threshold, which on the one hand enabled the detection of fraud but on the other hand led to an increased number of false positives. More importantly, purely unsupervised approaches are not interpretable and generally generalize poorly to unseen patterns. This suggests that while the model is valuable, in order to detect fraud, other techniques or decision systems

Future study

For the sake of further improvement of its efficacy and robustness, the model will be further investigated in future work on the following directions. Dynamic Thresholding: Take into account class specific error distribution, or percentile based cut-offs, and come up with dynamic thresholding strategies. SMOTE (Synthetic Minority Oversampling Technique) and ADASYN as Data Rebalancing Techniques are also applicable to more balance training datasets. Sew autoencoders with a classifier like Random Forest, XGBoost as an input feature to reduce error in reconstruction. SHAP or LIME should be investigated for explaining model predictions and enhancing stakeholders' trust. To improve the strength of the fraud signal: Adding transaction time windows, frequency patterns and a merchant level behavior. Adaptability and reliability: The model should be validated across different financial datasets to evaluate adaptability and reliability across different transaction environments.

References:

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, 90–113. https://www.sciencedirect.com/science/article/pii/S1084804516300571

Minastireanu, E.-A., & Mesnita, G. (2019). An analysis of the most used machine learning algorithms for online fraud detection. *Informatica Economica*, *23*(1). https://revistaie.ase.ro/content/89/01%20-%20minastireanu,%20mesnita.pdf

Viswanatha, V., Ramachandra, A. C., Deeksha, V., & Ranjitha, R. (2023). Online fraud detection using machine learning approach. *International Journal of Engineering and Management Research*, *13*(4), 45–57. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4533856