CAPSTONE PROJECT REPORT

ONLINE PAYMENT FRAUD DETECTION – ANALYSIS USING PYTHON MODELS

Abhinav Kumar Sangi, Keerthika Kanagaraj and

Sumesh Chakkaravarthi Purushothaman

College of Professional Studies, Northeastern University

ALY6140 Python & Analytics Systems Technology

Prof. Navdeep Singh Dhanjal

October 15, 2024

Introduction:

Essentially, the core of the Capstone Project would be based on analyzing the dataset for online payment transactions to detect fraudulent activities. As much as the digital economy is growing by leaps and bounds, there is also a corresponding increase in fraudulent activities that seriously put financial institutions and consumers at risk. Hence, the prime focus of the industry has fallen on fraud detection, especially since e-commerce has gained momentum. For this challenge, the questions listed below are considered in the project in an attempt to make a preliminary investigation of fraud detection. Namely, which of the transaction features is most indicative of fraudulent cases? Does it involve transaction amount or balance change? How does transaction type, transfer versus payment, relate to the fraud? And lastly, how does the sender/receiver side balance variation affect the case of fraud? Finally, we benchmark a number of machine learning model performances in order to find the best strategy for efficient identification of fraudulent transactions.

The following models will be implemented in the project: Linear Regression, Decision Tree Regression, and Random Forest Classification. We can implement Linear Regression to explore the relationship between the transaction amount and fraud probability. Continuing with the text examples, we shall apply Decision Tree Regression on the data to model nonlinear relationships and develop a fraud risk score for the transactions. In this regard, Random Forest Classification is an ensemble method applied to increase the predictiveness by merging numerous decision trees and emphasizing the most informative features concerning fraud. Answering research questions, this project evaluates models that may provide insight into fraud detection mechanisms useful to financial institutions by reducing risks related to online payments.

Explanatory Data Analysis

The data analyzed in this project consists of approximately 660,000 online payment transactions, all with a wide range of features related to fraud detection. Features include transaction type, such as transfer or cash-out, the amount of money in question, sender and receiver balance before and after the transaction, and a fraud flag on whether this was marked as fraudulent or not. This data provides a broad overview of online payment behavior and would hence be quite suitable for building fraud detection models. Rich variability in the feature space can well be exploited to explore in-depth patterns and correlations that could indicate fraud.

Extensive cleaning of data was done before actual predictive modeling to ensure accuracy in the results. Next would come imputation or removal of rows with missing values, depending on how important the feature in question is. Also, some numeric variables like transaction amounts and balances were normalized for consistency and better model performance. Besides, outlier detection was performed, and transactions with very extreme, apparently unrealistic values were removed to reduce the noise in the dataset. These steps helped maintain data integrity and ensured that model training was reliable.

Further, various visualizations were made for understanding data. The histogram of transaction amount showed that most of the transactions were of smaller amounts, whereas a few larger transactions were quite an exception. A heatmap of the correlation between variables showed that with respect to the fraud flag, the transaction amount and the balance features are moderately correlated. Furthermore, in fraud prevalence analysis by transaction type, it emerged that transfers and cash-out transactions are more predominantly related to fraud compared to other transaction

types. These exploratory analyses set the basis for feature engineering and informed our decisions when doing model selection and tuning.

Predictive Models

In this section, three models were built to detect fraudulent transactions: Linear Regression with Regularization, Decision Tree Regression, and Random Forest Classification. Each model was evaluated based on its performance metrics.

1. Linear Regression with Regularization

Since fraud detection can involve more intricate nonlinear relationships, Polynomial Features were used to model higher-order relationships between variables. For instance, we extended the linear model with squared terms or interaction terms between the features to fit the nonlinear nature of fraud data.

Further, we used the Lasso Regression algorithm, a type of linear regression with an added L1 regularization penalty. Later, we employed Lasso regression to perform shrinking overfitting by penalizing larger coefficients, and hence it actually conducted feature selection in this model. Because in this model, some of the feature coefficients are forced to be exactly zero, which also reduces model complexity and makes it more interpretable.

Key Parameters for Lasso Regression:

alpha: Regularization parameter. With an increase in the value of alpha, the estimates of coefficients will be forced closer to zero, thereby giving a more regularized model. We tuned alpha based on our implementation by doing a cross-validation in order to find an optimal balance between bias and variance.

max_iter: Maximum number of iterations taken for the convergence of the solver. We used max_iter = 1000 to converge the solution.

tol: Tolerance for optimization. To optimize better, we have used a lower value, viz., tol = 1e - 4.

The best Lasso Regression model was chosen using R^2 Score, which had an R^2 of 68%, thus providing a good baseline but failing to encapsulate the full complexity of the dataset.

2. Decision Tree Regression

Decision Trees are useful in fraud detection because they are able to capture nonlinear relationships between different features. For example, a transaction balance may vary based on type of transaction and previous behavior of customer with respect to fraud. These kinds of interactions are handled much better by Decision Trees than by any linear model.

Important Parameters of Decision Tree Regression:

max_depth: This is the maximum depth of this tree. It basically defines how deep the tree can grow and thus does not allow over-complexity. That could also allow the model to capture more information but might risk overfitting. After tuning, we set max_depth=15 to keep the tree flexible but not overly overcomplicated.

min_samples_split : The minimum number of samples required to split an internal node. Smaller values would prevent the tree from picking nodes that contain too few data points. This will reduce overfitting. We shall use min_samples_split=10.

min_samples_leaf: The minimum number of samples required to be at a leaf node. Putting min_samples_leaf=5 made the splits present some informative insights without making extremely small partitions in data.

These hyperparameters were then tuned with GridSearchCV to provide the best configuration of the trees. The Decision Tree model delivered a .015 MSE with an R^2 score of 75%. Further, it provided insight from the model in the form of a derived Fraud Risk Score where the risk of every transaction was quantified.

3. Random Forest Classification

It has been applied for fraud detection since Random Forest Classification demonstrates excellent performance with big datasets and high-dimensional feature spaces. Besides, it returns the feature importances, which can be used to learn which of the features, for instance, transaction amount and balance changes, are more indicative of fraud. This interpretability is of high importance to bring out those aspects of the transactions that contribute most towards fraudulent behavior.

Key Parameters for Random Forest Classification:

n_estimators: The number of trees in a forest. We used n_estimators=100, so we could have a good balance between computational efficiency and model performance. max_depth: The maximum depth of each tree. Then obviously, the deeper the tree may fit more patterns but then risks overfitting. We came to this result due to tuning. Thus, we set max_depth=12. min_samples_split: As explained earlier, in Decision Tree, it represents the minimum number of samples to consider creating a split. In this model we have used min_samples_split=10. max_features: This defines the number of features to consider at every split. We have used max_features='sqrt' - that means a number of features equal to the square root of the total number of features taken into consideration, will be considered at each split. This is done to keep diversity amongst the trees in the forest. Among these three, the Random Forest model performed the best, with an R^2 score of 80%. It also provided a better insight into which feature played an important role in determining fraud-for example, the amount transferred in the transaction and the balance before and after the transaction. In general, forests yield robust predictions with a low risk of overfitting and, therefore, are suitable for this problem.

Interpretations & Conclusions

Analysis through this project depicts that models using machine learning can be quite efficient in the detection of fraudulent online transactions. Among the different models used in the analysis, such as Linear Regression, Decision Tree Regression, and Random Forest Classification, the Random Forest model proved the best and most accurate among them to identify fraudulent activities. The ensemble approach of the Random Forest model basically aggregates the predictions of multiple decision trees that greatly reduce the risk of overfitting and enhance the reliability of prediction. It identified striking features of transaction amount and pre- and post-balance differences, noting that certain transaction types, such as cash withdrawal and transfers, were highly likely to be fraudulent.

The results highlight the need for efficient machine learning techniques in fraud detection systems of financial institutions. While Linear Regression had a place in providing some foundational understanding of how transaction amounts relate to fraud risk, it was really less effective compared with the more sophisticated tree-based models. This might form the basis for further research in the application of advanced models, such as Gradient Boosting Machines or Neural Networks, to better advance these detection capabilities and improve security for digital payment systems that are continuously changing.

References

Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Media.

Kaggle. (n.d.). Online Payment Fraud Detection Dataset. Retrieved from https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection

Scikit-learn Documentation. (n.d.). Ridge and Lasso Regression. Retrieved from https://scikit-learn.org/