Final Project: Credit Card Transaction Fraud Detection

Group 3: Zeyang Yu, Ziqi Lin, Pengyu Zhao, Lijia Yu

MSDS 422: Practical Machine Learning

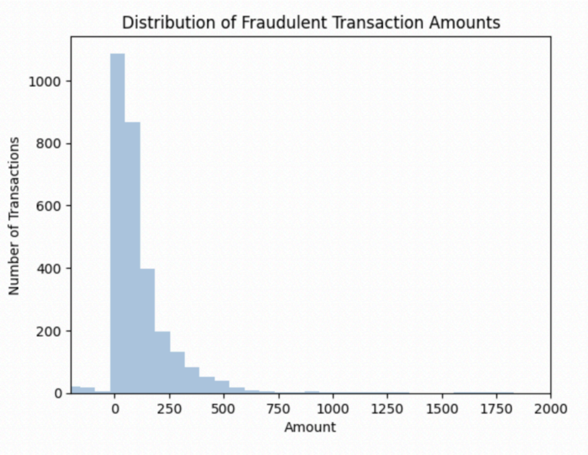
1. Background

Fraud detection has become a crucial issue across multiple sectors, including banking, insurance, law enforcement, and government organizations. In recent years, there has been an uptick in fraud cases, making the ability to differentiate between legitimate and fraudulent financial activities more important than ever. With the global shift towards digital transactions, the use of credit cards and online payment platforms has surged, leading to an increase in fraud occurrences. These fraudulent activities result in significant financial losses for both financial institutions and consumers. Therefore, we will conduct a thorough examination of the different techniques used for fraud detection.

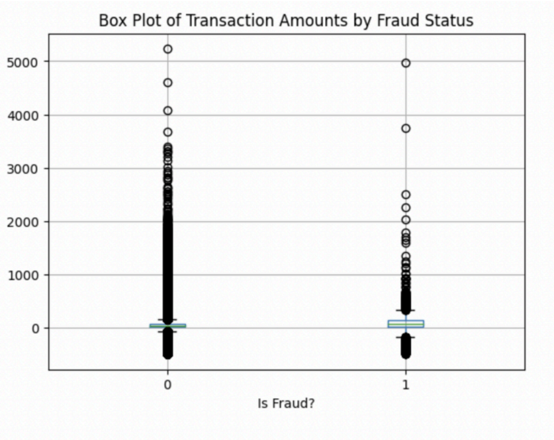
Our project's goal is to pinpoint fraudulent transactions within an imbalanced dataset (Kaggle dataset). Our strategy involves exploring and identifying the most effective algorithms capable of detecting fraud, including Support Vector Machines (SVMs), Random Forests, and Neural Networks. Additionally, we plan to fine-tune the optimal settings for these models, such as the learning rate and the number of epochs, to enhance their performance. Given the dataset's biased nature, we intend to implement strategies like undersampling and oversampling to address and mitigate the imbalance issue.

2. Exploratory Data Analysis:

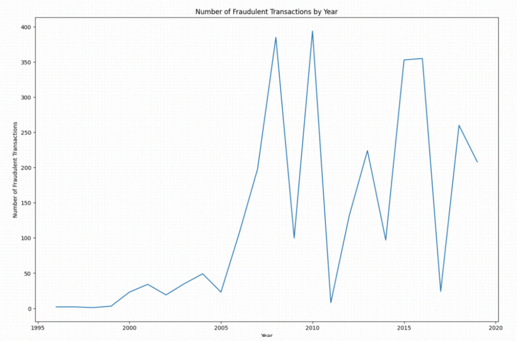
To explore the explanatory and predictive variables in the dataset, we conducted a visualization analysis. Firstly, we focus on fraudulent transaction amounts. In the distribution of fraudulent transaction amounts, we found the heavy skew toward lower transaction amounts could indicate that fraudsters prefer to keep transaction amounts small, perhaps to avoid

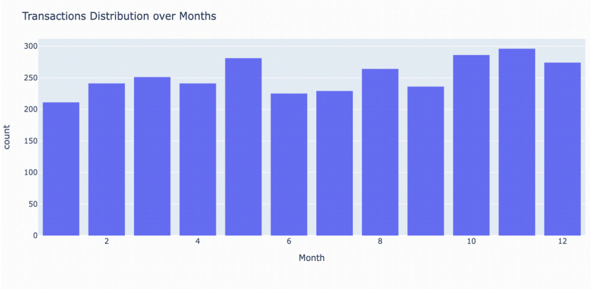


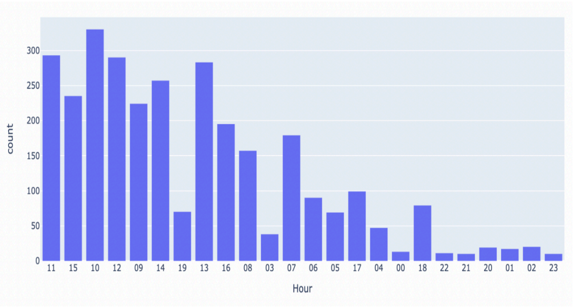
detection. High-value fraudulent transactions are significantly less common according to this distribution. Then, we made a boxplot to compare transaction accounts based on whether they were fraudulent or not. This boxplot indicates that while most transactions (fraudulent or not) are of a lower amount, there is a presence of high-value transactions that are outliers to the typical amounts. The slightly higher median for fraudulent transactions could imply that when fraud occurs, the transaction amounts tend to be higher than the median of non-fraudulent transactions.



We also explore fraudulent transactions in different time dimensions. From the line plot that shows the number of fraudulent transactions by year, we can see that the number of fraud cases had been growing steadily since 2005, peaking in 2007 and 2008 during the Great Recession and surging again in 2016. The histogram on the right does not present a significant difference in the distribution of fraud transactions by months. However, the number of fraudulent activities rose slightly towards the end of the year. The histogram of fraudulent transactions by hour of the day shows that most fraudulent activity occurred during the day.



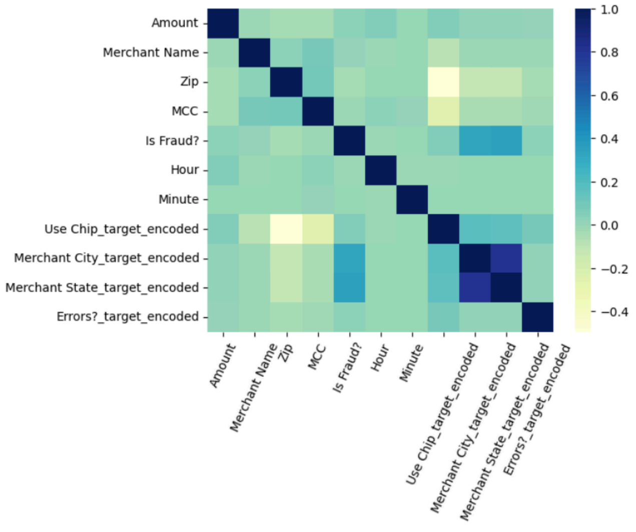




To investigate potential predictors, we created a correlation matrix to find out features that are highly correlated to the dependent variable 'Is Fraud?'. The matrix shows that the dependent variable does not have a strong relationship with other variables, while the categorical variable card type has a relatively high correlation with the variable card brand. The correlation matrix provides a clear picture of the relationships between variables in the fraud transaction detection dataset, which can be a reference for us to select appropriate variables in our model training stage.

3. Feature Engineering

In the data set, we transformed the ‘Time’ variable from a string representation into separate numerical or categorical columns for the hour and minute, making it easier to use in various analyses or machine learning models. Also, we extracted the day of the week and mapped it to its name.



We created a clean function designed to preprocess a pandas DataFrame df before it is used for machine learning training or prediction. First, it converts the data type of the 'Hour' column to float to ensure consistency in the data type, which is important for modeling.
Then, it scaled the "Amount" Column by using StandardScaler to standardize. This ensures that the 'Amount' feature will have a mean of 0 and a standard deviation of 1, helping to mitigate issues arising from features being on different scales. There exist a lot of categorical variables, so we applied binary encoding to the categorical variables. Binary encoding converts categorical variables into a series of binary columns, reducing the dimensionality compared to one-hot encoding while still capturing the presence of categories. This is particularly useful for categorical features with many levels. After encoding, the original categorical columns are dropped, and the new binary columns are added to the DataFrame. Finally, we converted all

columns in the data frame to float, so that all data is in a numerical format, which is necessary for most machine learning models.

4. Data Processing

4.1 Subset selection

In this data processing step, the dataset is meticulously prepared into two subsets to facilitate model building and performance evaluation. Initially, 10% of the data is randomly selected to create a subset specifically for model building. This selection process involves shuffling the original dataset to ensure diversity and representativeness. Subsequently, the dataset is reshuffled, using a different random seed, to generate a separate subset dedicated to testing the model's performance. By adopting this approach, the integrity and independence of the model development and testing phases are preserved, ensuring a reliable assessment of the model's generalization ability. This methodological rigor is crucial for objectively evaluating the predictive performance across unseen data, laying a solid foundation for robust fraud detection capabilities.

4.2 SMOTE and Dataset Partition

However, we found a significant class imbalance in our dataset, predominantly consisting of non-fraudulent transactions. In contrast, instances of fraudulent transactions are relatively rare, as evidenced by the overwhelming majority of non-fraudulent transactions (2,435,655) compared to the scant number of fraudulent transactions (3,035). This imbalance poses a substantial challenge to building effective machine learning models, as it can lead to a model biased towards predicting the majority class, overlooking the critical minority class of fraudulent transactions.

To address the class imbalance issue in our dataset, we have employed the Synthetic Minority Over-sampling Technique (SMOTE). This technique generates synthetic samples for the minority class to balance the class distribution. By introducing synthetic but plausible examples of fraudulent transactions, SMOTE enhances the diversity of our dataset and enables the machine learning model to learn a more balanced representation of both classes. This approach is crucial for improving the model's ability to accurately identify non-fraudulent and fraudulent transactions and generalize unseen data. By using SMOTE, we are taking a strategic step towards developing a more robust and effective fraud detection system.

We partitioned the resampled dataset into training and testing sets, allocating 80% for training and 20% for testing based on the balanced frame. This step is crucial as it ensures that the training and testing are conducted on distinct data sets. This practice is essential for evaluating the model's performance accurately. This systematic approach optimizes the numeric features for further machine learning tasks, setting the stage for more effective and efficient model training.

5. Methodology

We applied four machine learning algorithms—Logistic Regression, Naive Bayes, Random Forest, and Neural Network—to model fraud detection. Logistic Regression estimated the probability of transaction fraud based on a linear relationship between the input features. Despite its feature independence assumption, Naive Bayes effectively calculated the probability of fraud. The Random Forest algorithm, an ensemble of decision trees, offered robustness and accuracy without variable reduction. The Neural Network, through layers of interconnected neurons, excelled in identifying complex patterns indicative of fraud. To evaluate the models, we considered metrics like accuracy, precision, recall, F1 score, and AUC, mainly focusing on precision-recall trade-offs due to the high cost of misclassifications. Post-comprehensive performance assessments and hyperparameter tuning, our models demonstrated that machine learning is a formidable approach to detecting transactional fraud.

5.1 Logistic Regression

The logistic regression model employed for the task of identifying fraudulent transactions in your dataset has yielded great results. A deeper dive into the confusion matrix and the classification report reveals an impressive accuracy rate of 92%. Such a high accuracy level is commendable for a logistic regression model, which is relatively straightforward compared to more complex algorithms.

The F1-score, which harmonizes the precision and recall, stands at 0.91 for legitimate transactions and 0.92 for fraudulent ones. This symmetry in scores suggests that the model does not overly favor either class — it is equally adept at identifying legitimate and fraudulent transactions. A high precision rate of 0.94 for legitimate transactions means that when a transaction is labeled as non-fraudulent, it is correct 94% of the time. Meanwhile, the precision for fraudulent transactions is slightly lower at 0.89, suggesting a small margin for error where legitimate transactions might be incorrectly flagged as fraudulent. On the other hand, recall scores show a similar trend, with 0.89 for legitimate transactions and a slightly higher 0.94 for fraudulent transactions. This indicates that the model is slightly better at catching fraudulent transactions than in avoiding false alarms for legitimate ones.

The strong recall for fraudulent transactions is particularly important in the context of fraud detection, as failing to identify fraudulent activity can have significant financial implications. The fact that the model correctly identifies 94% of fraudulent transactions is encouraging. However, the number of false negatives, where fraudulent transactions are missed (28,762 instances), while relatively small in proportion to the dataset, still represents a potential area of concern. The existence of false positives (53,953 instances) also requires attention, as these instances can have a detrimental impact on customer experience and operational efficiency. Each false positive could mean a legitimate transaction is blocked or flagged for review, causing inconvenience to customers and potentially eroding trust in the system.

5.2 Naive Bayes

The Naive Bayes model generates a low F1-score of 73.94%, which suggests that the model has difficulty in achieving a balance between precision and recall. This could be because Naive Bayes assumes independence between features, which is not true in this credit card transaction dataset with correlated features. For instance, the two variables Transaction Amount and Merchant Category Code (MCC) could be correlated as certain MCCs may be associated with higher transaction amounts, and fraudulent transactions might occur more frequently in specific MCCs. Additionally, Naive Bayes might not be suitable for highly imbalanced datasets. While SMOTE may help balance the classes, it cannot guarantee perfect balance, especially in scenarios with extremely imbalanced data including our dataset of credit card transactions for fraud detection where the number of fraudulent transactions will be significantly lower than the number of non-fraudulent transactions. Naive Bayes could still struggle with class imbalance issues. This imbalance can cause the model to be biased towards class 0, which are the majority (non-fraudulent transactions), and poor at detecting class 1, which are the minority (potentially fraudulent transactions). The results also demonstrate that the Naive Bayes model generates high precision and recall for class 0 but relatively lower recall of only 44% for class 1. Because misclassifying fraudulent transactions can have significant financial implications, it is important to prioritize F1-score with indications of both precision and recall as the performance metric in our case. Therefore, Naive Bayes is not a suitable model for our project and we considered exploring other machine learning algorithms instead for transaction classification.

5.3 Random Forest

We performed hyperparameter optimization for a Random Forest classifier to detect fraudulent transactions. First, we defined a dictionary with the number of trees in the forest, the maximum depth of each tree, the function to measure the quality of a split, and the number of features to consider when looking for the best split. Then, we used Grid Search to identify the best combination of these parameters for model performance. With the best parameters found (using the 'gini' criterion, max_depth=20, max_features='sqrt', n_estimators=100), we trained the Random Forest classifier on the full training set with 3-fold cross-validation.

Based on the classification report, the model performed well, with precision scores of 1 for the class 0 (legitimate transactions), indicating that every transaction the model predicted as legitimate was indeed legitimate. The precision score of 0.99 for the class 1 (fraudulent transactions) indicated that 99% of transactions predicted as fraudulent were actually fraudulent. Meanwhile, recall scores of 0.99 for both classes meant that the model correctly identified 99% of all actual legitimate transactions and 100% of all fraudulent transactions.The F1-score for both classes was 0.99, indicating a strong balance between precision and recall. Overall, the accuracy of the model was 0.99, meaning that 99% of all predictions (fraudulent and legitimate) were correct. Based on this nearly 100% accuracy result, we believe that the Random Forest model has high predictive power.

5.4 Neural Network

Developing a Neural Network (NN) model for fraud detection aims to identify fraudulent transactions in a given dataset precisely. This particular NN model has three layers and employs

ReLU and sigmoid activations. It is optimized using the Adam optimizer and trained with a binary cross-entropy loss function. The model was trained over ten epochs with a batch size 128 on a Kaggle GPU. Its efficacy is demonstrated by its high recall of 0.9760, indicating its ability to detect a significant number of fraudulent transactions (475,944 true positives) and a precision of 0.8947, ensuring a low rate of false alarms (55,985 false positives). Additionally, the model demonstrates high accuracy (0.9305) and F1 score (0.9336), indicating its balanced performance in classifying fraudulent and legitimate transactions. Such a system is crucial for financial institutions as it can efficiently flag fraudulent activities, prevent economic losses, and protect consumer interests.

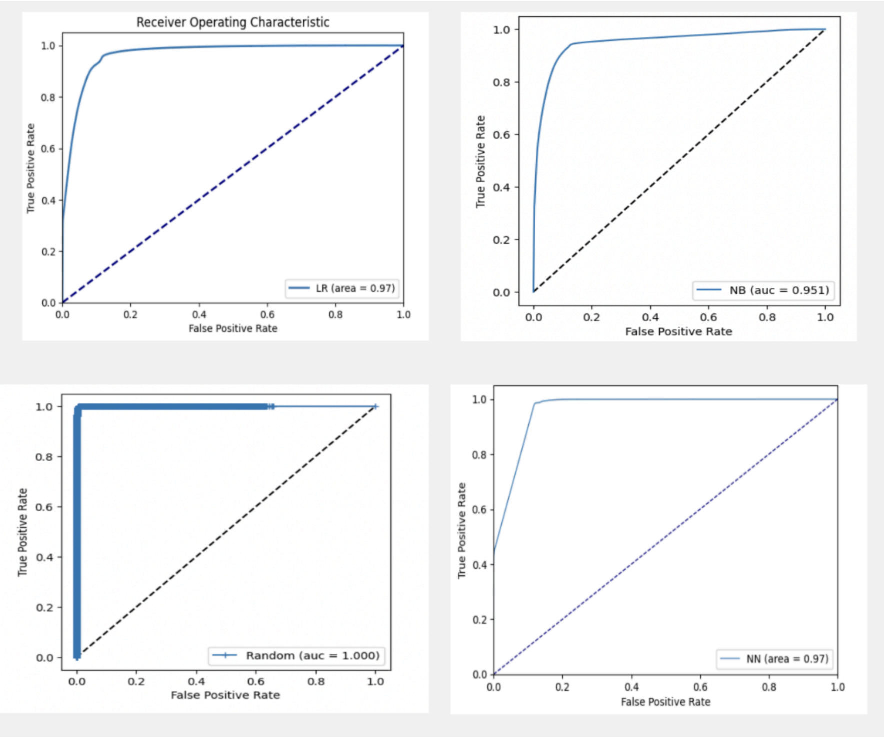
Moreover, we compared two neural network models, namely Model 1 and Model 2, based on several parameters such as the number of layers, epochs, accuracy, F1 score, loss, and training time per epoch. Model 1, when initially trained for 10 epochs, showed impressive performance with an accuracy of 0.9346 and an F1 score of 0.9367, indicating a well-balanced trade-off between precision and recall, with a relatively low loss of 0.1836 and an average training time of around 40 seconds per epoch. However, upon extending Model 1's training to 50 epochs, we observed a slight decrease in accuracy to 0.9314 and a minor increase in loss to 0.1919, which may suggest the possibility of overfitting despite the F1 score remaining robust at 0.9349. Additionally, the training time per epoch marginally increased to approximately 43 seconds. Model 2, which included four layers and was trained for 10 epochs, registered a lower accuracy of 0.9263 and an F1 score of 0.9292, along with a higher loss of 0.2185, and took about 46 seconds per epoch. This result may indicate that the increased complexity of Model 2 does not necessarily lead to better outcomes.

The study results indicate that adding more layers to a model does not necessarily lead to better performance, as demonstrated by the lower metrics achieved by Model 2 compared to the original Model 1. Additionally, overtraining the model (Model 1 Extended) can cause overfitting, as indicated by the decreased accuracy and increased loss. Therefore, finding an optimal balance between model complexity and training duration is crucial to achieving the best performance. The initial setup of Model 1 strikes a good balance between model complexity and training efficiency, resulting in high accuracy and F1 score with reasonable loss. This makes it an ideal choice for tasks like fraud detection, where precision and recall are crucial. These results highlight the importance of monitoring model performance metrics beyond accuracy, such as F1 score and loss, as well as training efficiency, to guide the model selection process for effective real-world applications.

6. Results

	Accuracy	**Precision**	**Recall**	**F1 Score**	**Training Time**
Naive Bayes	0.72	0.81	0.72	0.74	2.36 secs
Logistic Regression	0.92	0.92	0.92	0.92	42 secs
Random Forest	0.99	0.99	0.99	0.99	19.44 mins
Neural Network	0.93	0.89	0.98	0.93	40 secs per epoch

The Random Forest model significantly outperforms the other models and achieved the highest accuracy, precision, recall, and F1 score, with a near-perfect score of 0.99 in each but a longer training time of about 19.44 minutes. Logistic Regression and Neural Network also performed well. The Logistic Regression has scores of 0.92 across all metrics and a much faster training time of 42 seconds. Meanwhile, the Neural Network has high recall (0.98), F1 score (0.93), accuracy (0.93), and slightly lower precision (0.89). Besides, Naive Bayes has the lowest performance but is the fastest to train.



The ROC curves show that the Random Forest model achieves a perfect AUC of 1.00, indicating an exceptional ability to discriminate between classes. Additionally, the Logistic Regression and Neural Network models also achieved excellent performance with high AUC values of 0.97. The Naive Bayes model has a lower AUC of 0.95, which is still a good performance metric. Overall, these curves suggest that all models have good classification capabilities, with Random Forest being the most outstanding. However, the perfect AUC for Random Forest should raise an overfitting problem.

7. Conclusion

In conclusion, the exploration of different machine learning models for the detection of fraudulent transactions in the provided dataset has yielded informative insights into their respective performances and practical applications. The analysis covered Naive Bayes, Logistic Regression, Random Forest, and Neural Network models, each with its own strengths and limitations.

The Naive Bayes model, while the least accurate, offers simplicity and speed, making it an excellent candidate for initial benchmarking and scenarios where computational resources are limited. Despite its modest performance metrics, it serves as a valuable baseline from which improvements can be gauged.

Logistic Regression stands out for its strong performance balanced with efficiency. An accuracy of 92% and similar F1-scores for both classes demonstrate its robustness. Its relatively short training time and interpretability make Logistic Regression a compelling option for environments that require rapid decision-making capabilities. However, the false positives and negatives indicate room for improvement. It is recommended that further fine-tuning of the model be considered, possibly through enhanced feature engineering or regularization techniques to optimize its performance.

The Random Forest model exhibits outstanding accuracy, precision, recall, and F1-score, all rounding to 99%, and a perfect AUC of 1. Its performance is unrivaled in this evaluation, although its training time is significantly longer. For applications where accuracy is paramount and computational time is not a constraint, Random Forest is the recommended model. Nonetheless, practitioners should be cautious of the model's complexity and ensure that overfitting is addressed through proper validation techniques.

The Neural Network model presents a high recall rate, which is particularly desirable in fraud detection systems where failing to detect fraud can be costly. Its training time is reasonable, and with an accuracy and AUC of 93% and 97% respectively, it is a powerful contender. However, Neural Networks require extensive expertise to tune and interpret, making them less accessible for smaller teams or those without deep technical knowledge.

As a final recommendation, when selecting a model for deployment, one must consider not only the performance metrics but also the specific needs of the application, including interpretability, computational resources, and the cost of errors. A model that balances these factors with high performance would be ideal. In certain cases, an ensemble of models could be employed to leverage the strengths of each.

In summary, while each model has its own set of advantages, the Logistic Regression model, with its notable balance of accuracy and efficiency, may offer the best trade-off for many practical applications in fraud detection. However, for the most critical applications where the highest accuracy is required, the Random Forest model is the clear winner, given that the trade-off in training time is acceptable.

References

"Credit Card Transactions." 2021. Kaggle. October 14, 2021. https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions.

Maniraj, S P, Aditya Saini, Shadab Ahmed, and Swarna Deep Sarkar. 2019. "Credit Card Fraud Detection Using Machine Learning and Data Science." *International Journal of Engineering Research and Technology* 08 (09). https://doi.org/10.17577/ijertv8is090031.

Rahman, Md Jahidur, and Hongtao Zhu. 2023. "Predicting Accounting Fraud Using Imbalanced Ensemble Learning Classifiers – Evidence From China." *Accounting & Finance* 63 (3): 3455–86. https://doi.org/10.1111/acfi.13044.

GitHub link

https://github.com/kathyyu14/MSDS-422-Final-Project