STAT 536 - Case Study 5

Credit Card Fraud

Sam Lee, Paul Crowley

Fraud detection is an important function of banking and serves to protect both the bank and the client. The vast majority of credit card transactions are valid, and this imbalance makes it difficult to detect fraudulent transactions. However, it is still essential to identify real cases of fraud without flagging clients for valid everyday purchases. Using dimension-reduced credit transaction data from the bank, we build a random forest model tuned with optimal hyperparameters and trained on added synthetic fraudulent data to correctly identify real fraudulent transactions 99.9% of the time while minimizing false negative fraudulent cases.

Introduction

Credit card fraud represents a significant financial burden, with estimated global losses reaching approximately $22 billion annually. In response, credit card companies employ advanced machine learning techniques to identify fraudulent transactions accurately. This report examines a dataset containing around 300,000 credit card transactions, of which only 492 are fraudulent, translating to a prevalence of approximately 0.1%. Given the rare nature of fraudulent events, our objective is to develop a high-performing model that can detect fraud with precision, minimizing both financial losses and false positives, which could otherwise disrupt legitimate users.

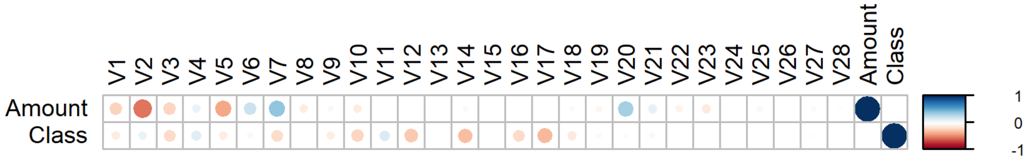


Figure 1: Correlation matrix between factors used in model specification. Note that all partial components (V1-V28) are, by construction, uncorrelated with each other.

Several potential challenges in the data may impact our analysis. Given the nature of the data, we will estimate a series of non-parametric binary classification models to predict whether a transcaction is fraudulent. The dataset's extreme class imbalance raises concerns about model performance, especially with a tendency to misclassify rare fraudulent transactions. Additionally, while principal components enable dimensionality reduction, they also reduce interpretability since they lack direct transactional meaning—while we ignore this caveat in this analysis to prioritize prediction, the lack of interpretability means we are unable to come up with an *a priori* non-linear specification for a parametric model to account for any non-linear trends in the data. Since these trends are unknown, we rely on non-parametric specifications that will better be able to capture unique interactions and non-linear effects. Each Pricinpal component, by construction, is uncorrelated with each other; however, this doesn't negate the possibility of *Amount* being correlated with any of the partial components (See Figure 1). In our analysis, we will evoke methods that are robust to multicolinearity as well as methods that are resilient to factors that aren't significantly meaningful.

A preliminary analysis reveals that as *Amount* increases, fraud is more likely (See Figure 2). However, we caution as interpreting this result as causal as there may be confounding effects unadjusted for. We also

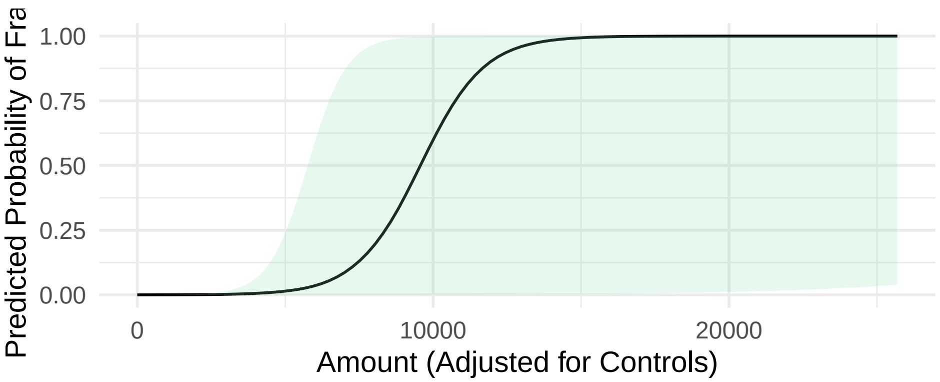


Figure 2: Relationship between transaction amount and likelihood of fraud. Estimates given by a logistic regression model with $n = 284,802$. Uncertainty estimates computed as the 95% quantile interval of $B = 100$ bootstrapped estimations.

acknowledge the potential non-linearity in *Amount* with respect to the likelihood of fraud (as captured by the uncertainty bounds in Figure 2).

Methodology

To identify fraudulent credit card transactions effectively, we consider two robust ensemble methods: Random Forest and Gradient Boosting. Both methods have demonstrated high predictive performance in classification tasks, particularly with imbalanced datasets, making them suitable for our analysis.

1. Using a Random Forest (RF) model will help us eliminate potentially unnecessary factors that would otherwise fit noise in the data through *bagging* (we describe the bagging method, as implemented through a RF model in the Appendex; see Algorithm 1). The RF model, by construction, will reduce the variance in our predictions. Additionally, since the RF model is an aggregation of decision trees, it will also help model the non-linear trends and complex interactions within our data. However, the extreme class imbalance may lead the RF model to underperform in identifying fraudulent transactions. In order to minimize this bias, we will tune fitting hyperparameters.

2. Similar to the Random Forest Model, Gradient Boosting (GB) uses a series of decision trees—by construction, this will help us model the complex and non-linear relationships in our data. However, in contrast from RF, rather than an *aggregation* of indenpent decision trees fit on a random selection of the data, GB builds trees sequentially, each one focused on correcting the errors of its predecessor. In each iteration, the model minimizes a differentiable loss function by adding a new tree that fits the residuals of the combined ensemble from previous iterations. For binary classification, we specify the Gradient Boosting model as[1],

$$\hat{f}_{GB}(x) = \sum_{m=1}^{M} \alpha_m h^{(m)}(x) \tag{1}$$

Where each new tree, $h^{(m)}(x)$, is trained[2] on the residuals (error) of the current prediction ($\hat{f}_{m-1}$). Similar to the RF model, the GB model yields high accuracy, models complex patters, and through hyperparameter tuning, allows for control of model complexity. The ability to "learn" from it's mistakes allows it to reduce

[1] Here we define M as the total number of trees, $h^{(m)}(x)$ is the m-th decision tree in the sequence, and α_m is the learning rate to control the contribution of each tree.

[2] We evaluate how well the new tree fits on the residuals of the current predictions by computing the negative gradient of the loss function, $L(y, \hat{f}(x)) = -[y \log \hat{p}(x) + (1-y)\log(1-\hat{p}(x))]$ (that is using the binary cross-entropy loss for binary

bias iteratively, although we caution against overfitting a model like GB due to its inherent decision tree structure and sensitivity to model hyperparameters.

Model Evaluation

In the context of this problem, the criticality of accurately classifying credit card transactions cannot be overstated. Given the consequences, we consider false negative predictions—failing to identify fraudulent transactions—far more severe than false positives. To address this, we prioritize higher sensitivity over specificity, erring on the side of identifying potential fraud at the cost of increased false alarms[3].

Both the random forest and gradient boosting models acheived near perfect results during in-sample training with added synthetic minority class data in all performance metrics evaluated, but our primary focus was on the test predictions. The random forest model had very high sensitivity relative to the gradient boosting model, which is perhaps the most important consideration because it is more important to correctly identify fraudulent transactions than incorrectly identify valid transactions. The random forest also achieved much better positive predictive value, indicating that a higher proportion of predicted fraudulent transactions are actually fraudulent.

Table 1: Performance Metrics for Random Forest and XGBoost Models

Metric	**In-Sample**		**Out-of-Sample**	
	Random Forest	**XGBoost**	**Random Forest**	**XGBoost**
Sensitivity	0.9999	0.9960	0.9997	0.9952
Specificity	1.0000	0.9834	0.8000	0.8690
PPV	1.0000	0.9836	0.9997	0.9998
NPV	0.9999	0.9959	0.8286	0.2364
F2 Score	0.9999	0.9935	0.9997	0.9961
Accuracy	0.9999	0.9897	0.9994	0.9950

Given the reduced dimensions of the data that respect data privacy, interpretability was not much of a consideration in the model selection process. If a highly interpretable model like logistic regression were chosen for this fraud classification task, interpretations could only be made about the principal components rather than the the original variables. Thus, both proposed models, are designed with predictive performance in mind. Ultimately, the random forest model produced more desirable results for test predictions, and this model was chosen to accomplish research objectives.

The random forest model is specified below with hyperparameters selected through a randomized grid search[4] (See **?@eq-rf** in the Appendix for parameterization) and performance evaluated by $F2$ score to prioiritize high sensitivity over high specificity. The random forest model does not operate on any distributional assumptions. The model results do not depend on any such justifications. All features were kept in the model to maximize predictive performance, but each tree only classifies based on 4 features to ensure unique contributions to the ensemble.

classification),

$$g_i(m) = -\frac{\partial L(y_i, \hat{f}(x_i))}{\partial \hat{f}(x_i)}\bigg|_{\hat{f}(x)=\hat{f}_{m-1}(x)}$$

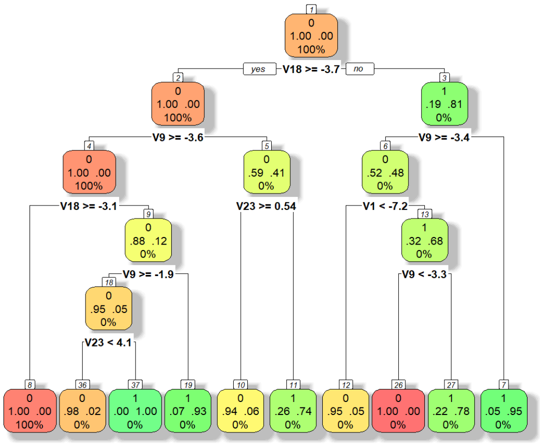
These often-called ”pseudo-residuals” (the vector of $g_i(m)$) is what the subsequent model is trained on.

[3] As such, we evaluate our models using the $F_{\beta=2}$-Score, that is, using $F_\beta = \frac{(1+\beta^2)TP}{(1+\beta^2)TP+\beta^2 FN+FP}$, where *TP, FN,* and *FP* represent the number of true positive, false negative, and false positive predictions committed by the model, respectively, where we set $\beta = 2$.

[4] We used a 70-30 train-test split for cross-validation on the tuning parameters, using the out-of-sample $F2$ score as the validation criteria.

Results

As an ensemble model, it is not possible to visualize the complete operations of the random forest. However, individual trees used in the ensemble can easily be extracted to demonstrate its iterative process. The plot to the right shows where one such tree split the data with predicted probabilities for validity and fraud at each step.



(a) Individual Decision Tree Example

The random forest model does not provide interpretable parameters to analyze the impact of each partial component. However, metrics of variable importance and partial dependence plots can still be used to identify the most impactful variables. By permutating each partial component to eliminate any existing relationship with fraud, the decrease in accuracy can be compared to determine which partial components have the greatest effect on fraud. Partial dependence plots demonstrate how the predicted probability of fraud changes as the partial component increases. These plots are given below.

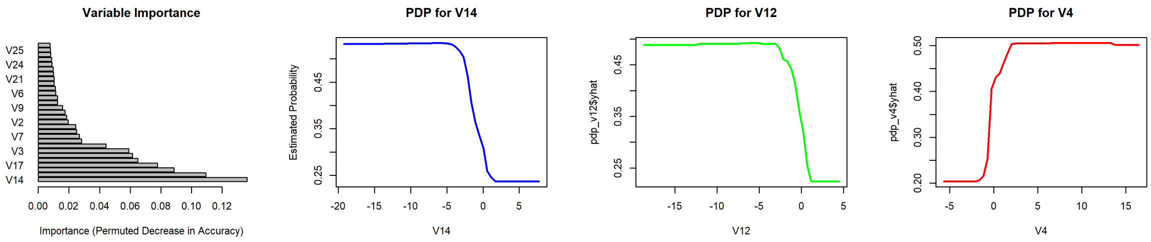


Figure 4: Variable Importance and Partial Dependence for Most Impactful Components

The random forest model does an excellent job identifying the fraudulent transactions. Given a transaction is fraudulent, the model can accurately detect it 99.99% of the time in tested cases. Of the five new transactions for which we do not know whether they are valid or fraudulent, the model predicts only the third one to be fraudulent (See Table 2). Based on the model's sensitivity, it seems very unlikely that any of the other cases are fraudulent. This model can predict fraud with very high accuracy and emphasizes correctly predicting fraudulent transactions over correctly predicting valid transactions due to the high risks associated with fraud. The model is very complex, but its process is easily visualized, and important variables are also identifiable.

Conclusion

The analysis used a random forest model to predict fraud with great success. The model incorporated synthetic minority class data to counteract the drastic class imbalance used tuned hyperparameters to optimize predictive performance without overfitting. Given a fraudulent transaction, the model can identify it as such about 99.99% of the time. Of the 5 new transactions, the model predicted only the third one to be fraud. However, certain shortcomings were observed: Random Forest exhibited lower predictive accuracy given a valid transaction, leading to a higher rate of false positives. These results highlight the trade-off between sensitivity and specificity in imbalanced datasets. Additionally, the limited interpretability of the models could impede actionable insights to prevent fraud. To address these limitations, future work explore anomaly detection approaches to improve specificity and utilize interpretability tools such as SHAP to provide greater transparency.

Appendix

Algorithm 1 Random Forest Algorithm

```
Input: Training data with $n$ samples and $p$ features
Parameters: Number of trees $B$, number of features to consider at each split $m$ (where $m < p$)
for $b = 1, \ldots, B$ do
    Draw a bootstrapped sample of size $n$ from the training data
    Grow a decision tree $\mathcal{T}_b$ on this sample:
        1. At each node, randomly select $m$ features from the $p$ available features
        2. Split on the best feature among the $m$ chosen features (based on some criterion, e.g., Gini impurity for classification)
        3. Repeat until the stopping criterion is met (e.g., maximum depth or minimum node size)
end for
Prediction: For a new observation $x_0$
    1. For each tree $b = 1, \ldots, B$, obtain a prediction $\hat{y}^b(x_0)$ by passing $x_0$ down tree $\mathcal{T}_b$
    2. For regression: average the predictions:
           $$\hat{y}(x_0) = \frac{1}{B}\sum_{b=1}^{B}\hat{y}^b(x_0)$$
    3. For classification: take the majority vote:
           $$\hat{y}(x_0) = \mathrm{mode}(\hat{y}^1(x_0), \ldots, \hat{y}^B(x_0))$$
```

$$\hat{y}(x_0) = \begin{cases} 1 & \text{if } \frac{1}{B}\sum_{b=1}^{B}\hat{y}^b(x_0) \geq T, \\ 0 & \text{otherwise.} \end{cases}$$

Where:

$$\begin{aligned} B &= 200 \quad \text{(Number of trees)}, \\ m &= 4 \quad \text{(Number of features at each split)}, \\ S &= 1 \quad \text{(Minimum node size)}, \\ f &= 0.6 \quad \text{(Proportion of samples used for each bootstrap)}, \\ D &= 15 \quad \text{(Maximum depth of each tree)}, \\ T &= 0.46 \quad \text{(Probability threshold for classification)}. \end{aligned}$$

Table 2: Predicted fraudulent cases among unkown test cases
(partial components ommitted for brevity)

Case	Fraud	Amount
1	No	0.89
2	No	3.84
3	Yes	1.00
4	No	1.00
5	No	180.40