Automating Credit Card Fraud Detection with Machine Learning

Samuel Showalter, Dr. Zhixin Wu | DePauw University

Credit Card Fraud Overview

Credit card fraud has spread rapidly in the past decade, fueled partially by the proliferation of online purchasing. This study explores optimal processes for automating fraud detection with machine learning algorithms and alternative sampling methods.

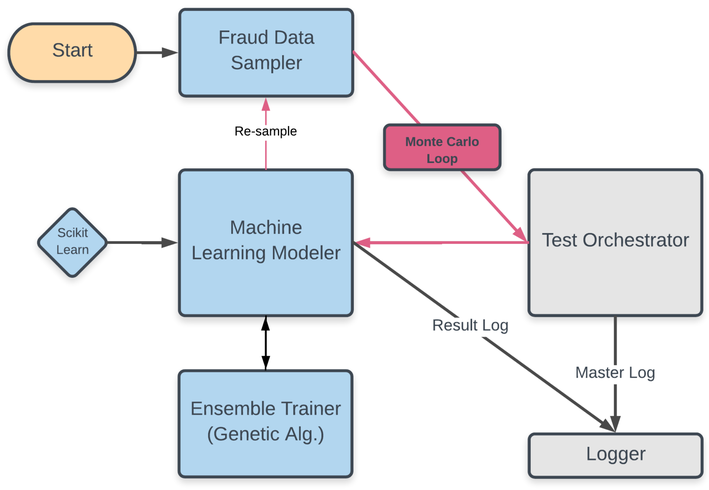
Challenges of Fraud Detection

- *High Societal Costs:* Fraud costs society ~$2.40 per $1 lost.
- *Data Imbalance:* ~0.2% of credit card data is fraudulent.
- *Concept Drift:* Purchasing behavior changes over time.
- *Verification Latency:* Humans must verify if a transaction is fraud.

Experimental Design

This study includes three considerations. First, what sampling methods most effectively train Fraud Detection System (FDS) algorithms? Two sample types – undersampling and Synthetic Minority Oversampling (SMOTE) – are tested. Second, an ensemble algorithm comprised of individual FDSs is trained with a genetic algorithm to attempt to outperform its components. Most importantly, we examine the relationship between traditional performance metrics (accuracy, F_1 Score) to cost-based metrics that calculate the cost of fraud directly. Firms may be losing revenue if the metrics they use to score their fraud systems are uncorrelated with actual cost savings. Logistic Regression (LOG), Linear SVC, Random Forest (RF), K-Nearest Neighbors (KNN), and Bayesian (GNB) classifiers were tested. Experiments were automated by a scalable fraud detection and documentation system.

Automated Fraud Detection System



Traditional v. Cost Performance

Prediction \ Actual Transaction	Fraud	Not Fraud
Fraud	True Positive (TP)	False Positive (FP)
Not Fraud	False Negative (FN)	True Negative (TN)

$$F_1 = 2\left(\frac{PPV * TPR}{PPV + TPR}\right) = \frac{2TP}{2TP + FP + FN}$$

The F_1 Score, calculated above, is considered a holistic measure of performance for imbalanced data. It merges *Positive Predictive Value (PPV)* and *True Positive Rate (TPR).*

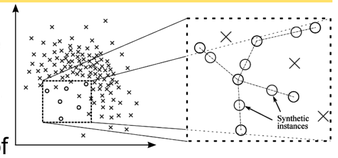
Conversely, fraud cost is derived from the matrix to the left, shown below. T_c, the cost of a transaction, is subtracted if it is fraudulent and prevented. If not prevented, the cost is increased by the fraud multiplier F_m *($2.40).* Time loss is given by C_e = *$1,* $C_f = C_l$ = *$10.* They are the cost of error, fraud, and loss.

Prediction Cost \ Actual Transaction Cost	Fraud	Not Fraud
Fraud	$-T_c + C_f$	C_e
Not Fraud	$F_m(T_c) + C_l$	0

$$F_c = (TP + FN)C_{fl} + (FP)C_e + \sum_{i=1}^{n} F_m(T_{ic}|T_{i=FN}) - (T_{ic}|T_{i=TP})$$

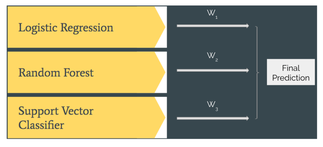
Data and Sampling Methods

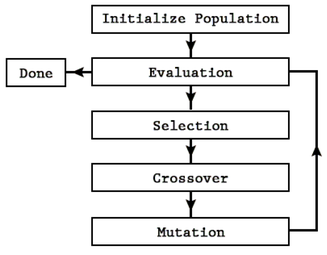
Our dataset is ~300k records, of which 492 were fraud transactions. They are from an unnamed European bank. Undersampling is conducted by randomly selecting a subset of the majority class to be combined with all records of the minority class, at a specified ratio. It seeks to emphasize the importance of minority classification.



SMOTE also selects a subset of the majority class, but then it embellishes the minority class by creating synthetic records. It does this by selecting a record and finding its most similar "neighbor." Traits for these two records are differenced, and a scalar from $U \sim (0,1]$ is applied to each difference before the whole sum is added back to the original component. This effectively creates a new fraud record.

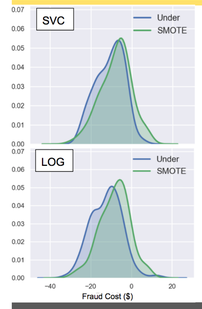
Genetic Algorithm Optimization





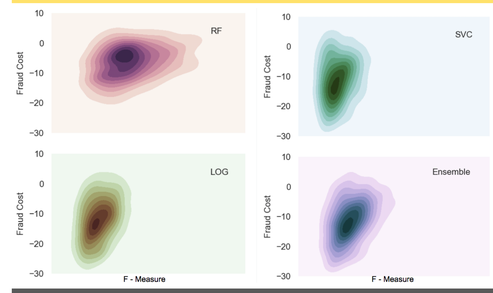
The best three models, shown to the left, are combined in a voting ensemble system. Together, each individual prediction is combined as a weighted average. To optimize these weights, a genetic algorithm is implemented with the structure diagrammed to the left. Evaluation is defined as the total cost of fraud F_c, and the probability of succession is determined accordingly. Mutation and crossover are conducted with bit manipulation using the binary form of each integer weight.

Sample and Model Performance



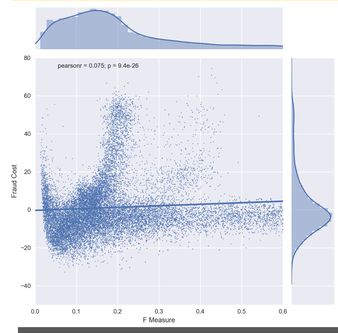
Undersamping outperformed SMOTE, and the best three models are shown to the left. A fraud ratio of 0.3 in samples was optimal. LOG performed best, in front of Linear SVC and Random Forest. KNN left out as the control.

Ensemble Results



While the ensemble had the second best performance out of all models, it did not predict fraud better than LOG. However, its performance was more consistent than LOG, showing promise for further research.

Evaluation Method Results



Surprisingly, the F_1 Score was not correlated with cost-based performance. This has potentially serious implications if cost matrices are not considered in a business setting. Companies may be losing money to fraud and not knowing it. Even AUPRC, one of the most sophisticated metrics for scoring fraud detection algorithms, does not weight fraud proportional to the cost it bears on society and firms.

Discussion

As blockchain and other online payment technologies arise, systems will be unprepared to detect fraud unless they are robust under uncertainty. This study shows undersampling outperforms SMOTE at training FDSs by re-balancing data to emphasize fraud over the majority class (non-fraudulent transactions). Moreover, we find promising evidence that ensemble classifiers can thrive in detecting fraud even when little contextual information exists. Lastly, we underscore the schism between statistical and cost evaluation. Statistics provide an objective standard from which models may be compared. However, cost performance is central to company success.

Acknowledgements: I wish to thank my advisor, Dr. Zhixin Wu, for allowing me this amazing opportunity to explore credit card fraud. Her mentorship made this project possible. I would also like to thank Dr. Chad Byers and Dr. Brian Howard for assisting me in the software development phase of this project.