**Fraud Detection in Online Transactions for Fastmeal by Tradeet**

## Introduction

Fastmeal by Tradeet is a new food delivery startup that connects restaurants with customers through a seamless ordering platform. Like many digital-first businesses, Fastmeal faces the challenge of fraudulent transactions, such as stolen card use, chargebacks, and account takeovers. To protect the business and customers, a predictive system was developed to flag transactions likely to be fraudulent.

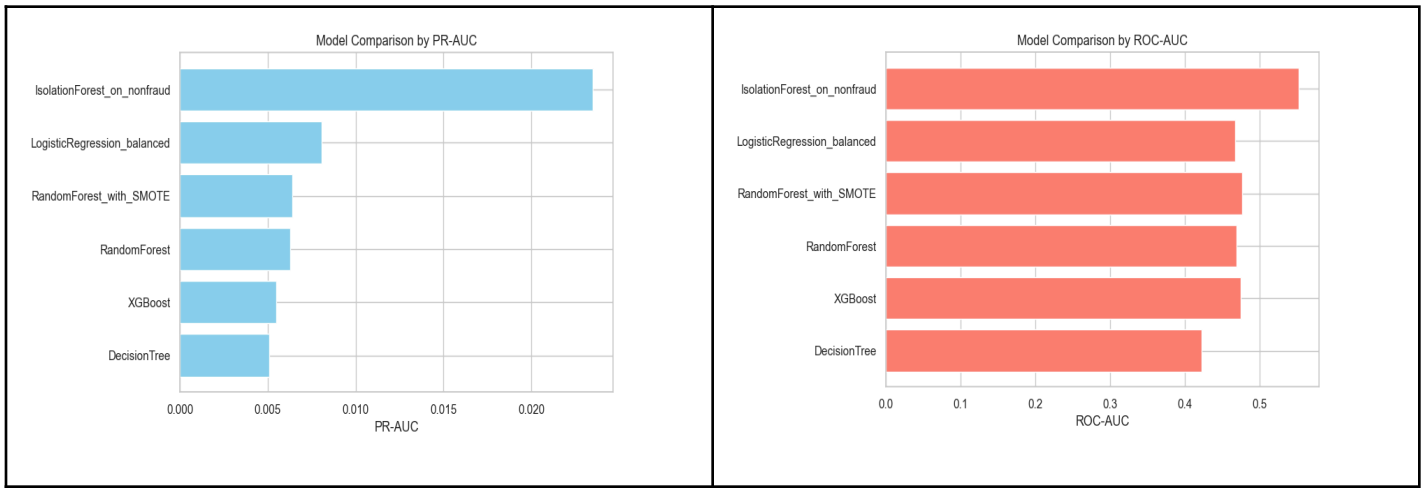
Each transaction record contains features such as order amount, payment method, device type, IP region, and whether the billing and delivery addresses match. Because real financial data could not be used, a synthetic dataset was generated with Gretel AI to preserve the statistical relationships and behavioral dynamics typically found in real transactions. Check these links for the synthetic data report and python notebook

## Approach

The modeling pipeline started with exploratory analysis, feature engineering, and preprocessing. Fraud detection is notoriously difficult due to class imbalance where genuine transactions far outnumber fraudulent ones. As emphasized in "Imbalanced Classification in Fraud Detection" (Data Reply, 2020), standard accuracy is misleading here. A model that predicts "not fraud" for every case would score high accuracy but be useless.

Therefore, the evaluation focused on Precision-Recall AUC (PR-AUC), which reflects how well a model detects the minority fraud class without generating excessive false positives, alongside ROC-AUC for robustness. Class-balancing strategies (undersampling, SMOTE, and class weights) were applied to test their effect on performance.

## Model Experiments and Results




## Discussion of Results

The results highlight a clear gap between anomaly detection and supervised classification. Isolation Forest, despite being unsupervised, achieved the highest PR-AUC (0.023), which is three to four times better than any supervised model. Its ROC-AUC of 0.55 is only slightly above random, but in highly imbalanced problems like fraud detection, PR-AUC is the more reliable metric. This suggests that anomaly detection can capture some of the hidden irregularities in fraudulent transactions, even when labeled fraud cases are extremely scarce.

Supervised models, including Logistic Regression, Random Forest (with and without SMOTE), XGBoost, and Decision Tree, all collapsed to near-chance performance with PR-AUC values between 0.005 and 0.008. This weakness is explained by the extreme class imbalance, the limited feature set (mainly numeric), and the ineffectiveness of SMOTE, which likely introduced noise instead of useful synthetic signals. Logistic Regression with class balancing performed slightly better than the other supervised models, showing the benefit of re-weighting. Random Forests did not gain much from SMOTE, XGBoost underperformed due to lack of tuning and weak features, and Decision Tree ranked lowest, reflecting its tendency to overfit in skewed datasets.

Overall, these findings confirm that anomaly detection is more effective than supervised learning in early fraud detection systems with limited features and few fraud labels. The results also underline the importance of using PR-AUC rather than ROC-AUC in evaluating imbalanced problems, since ROC curves can mask poor detection of the minority class. Moving forward, richer behavioral and categorical features will be essential for supervised models to match or surpass anomaly detection approaches.

**Assumptions, Key Decisions, and Trade-offs**

This project assumed that the synthetic data faithfully represented real Fastmeal transactions. The decision to prioritize PR-AUC over accuracy was deliberate, reflecting the business need to catch fraud even at the cost of some false alarms.

Key trade-offs included balancing interpretability vs. performance. Isolation Forest showed the best ranking score but is less explainable to stakeholders, while Logistic Regression was weaker but transparent. Similarly, oversampling increased recall but risked introducing artificial, unrealistic data patterns.

Data drift was not implemented in this experiment. However, it is crucial in fraud detection because fraud patterns and even customer behaviors evolve over time. For example, as Fastmeal grows, customers may make larger orders, use new payment methods, or transact from new regions. A model trained on older patterns could begin missing fraud or over-flagging legitimate activity. In production, drift detection could be handled with techniques like Population Stability Index (PSI) or Kolmogorov-Smirnov (KS) tests

**Recommendations and Conclusion**

Fastmeal by Tradeet should adopt a hybrid approach, combining Isolation Forest for anomaly detection with Logistic Regression or Random Forest for interpretability. To improve performance, richer behavioral features (e.g., transaction velocity, device fingerprinting, geographic checks) should be added. Continuous data drift monitoring is essential as customer and fraud patterns evolve. Imbalance handling techniques beyond SMOTE, such as cost-sensitive learning, should be explored. Finally, the fraud model should balance fraud prevention with customer experience by aligning tolerance for false positives with business priorities.

This assessment designed a fraud detection system tailored for Fastmeal by Tradeet. By testing anomaly detection, linear models, and tree-based ensembles, the study highlighted that fraud detection success depends less on picking the “fanciest” algorithm and more on addressing imbalance, selecting rich features, and monitoring drift.

Isolation Forest emerged as the best-performing model, showing promise for detecting rare fraud cases, while Logistic Regression and Random Forest remain valuable for transparency and stability. With further tuning, feature engineering, and a strong drift-monitoring strategy, Fastmeal can build a scalable fraud detection system that protects its growth and customer trust.

**This project emphasizes a core lesson that the handling of data imbalance and feature richness matters more than algorithm choice alone.**