Credit Card Fraud Detection

By Katelyn McDonald

1. Introduction

Credit card fraud poses a significant challenge for financial institutions as the volume of digital transactions continues to rise. Detecting fraudulent behavior is difficult not only because fraudsters constantly alter their tactics but also because fraudulent events make up a very small fraction of all transactions. Machine learning provides an effective framework for identifying unusual or suspicious activity by learning patterns that distinguish legitimate purchases from fraudulent ones. This project applies machine learning techniques to a large, highly imbalanced transaction dataset in order to develop models capable of identifying fraudulent behavior with strong reliability and practical deployability.

2. Dataset and Problem Structure

The dataset consists of more than one million transactions generated by the Sparkov credit card simulator. Each observation includes demographic characteristics, merchant information, geographic location, transaction attributes, and a binary label indicating fraud. Fraud accounts for roughly 0.58 percent of all transactions, making this a severely imbalanced classification problem. This imbalance makes simple accuracy ineffective because a model could achieve over 99 percent accuracy simply by predicting every transaction as legitimate. The dataset includes both numerical and categorical variables. Numerical features, such as transaction amount, timestamp, latitude, and longitude, provide quantitative information. Categorical features, including job category, gender, and merchant type, contain important behavioral signals but must be encoded appropriately for use in machine learning models. The data contains no missing values.

3. Preprocessing and Feature Engineering

All numerical variables were standardized using z-score scaling. Standardization ensures that algorithms sensitive to feature magnitude, such as logistic regression, behave appropriately without unintentionally weighting certain variables more heavily. Categorical features were transformed using one-hot encoding, which converts each category into a binary indicator without imposing a false numerical order. This is particularly important for fraud detection, where categories such as “merchant type” or “job” carry meaning but have no natural ranking. Several engineered features were introduced to capture behavioral patterns commonly associated with fraud. The transaction hour was extracted from the timestamp, and transactions occurring late at night were flagged due to their higher risk. The geographic distance between the customer and the merchant was calculated using their latitude and longitude coordinates since purchases made far from the customer’s typical location can be an indicator of fraud. These engineered features helped improve model performance.

4. Class Imbalance Handling

Because fraudulent transactions are extremely rare, the training data was balanced using the Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates new synthetic minority examples by interpolating between existing fraud cases, allowing the models to learn decision boundaries that better distinguish rare fraudulent behavior. SMOTE was applied only to the training split.

5. Model Development

The two models developed based on their suitability for real-world fraud detection systems were Logistic Regression and Random Forest. Logistic Regression provides a simple, interpretable baseline model that performs well on large datasets and supports real-time scoring due to its computational efficiency. Random Forest, a tree-based method, captures nonlinear relationships and interactions between features, offering more expressive modeling power. A stratified sample of 200,000 transactions was selected from the original dataset to reduce computational demands while maintaining class proportions. This sample was split into training and testing sets using a stratified 70/30 partition. SMOTE was applied to the training partition only. Both models were then trained on the balanced training data and evaluated on the untouched test set.

6. Evaluation Metrics

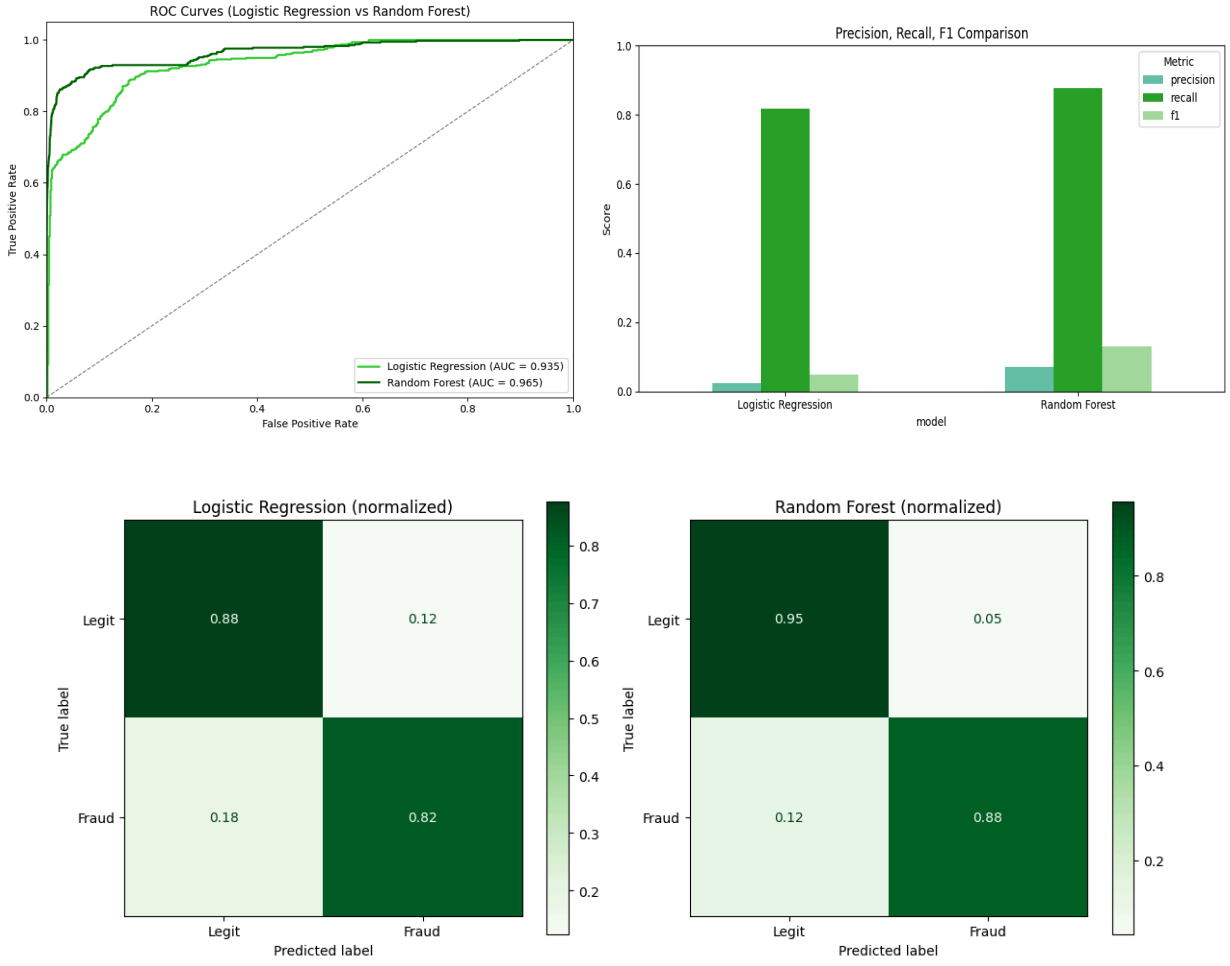
Because accuracy is not informative in imbalanced problems, model performance was assessed using recall, precision, F1 score, and ROC AUC. Recall measures the proportion of fraudulent transactions correctly identified, and is important since missing a fraud can be extremely costly. Precision indicates how often the model's fraud predictions are correct, reflecting the tradeoff between identifying fraud and avoiding excessive false alarms. The F1 score combines precision and recall into a single metric, and ROC AUC evaluates the model's ability to rank fraudulent transactions above legitimate ones across all possible decision thresholds.

7. Results

Both models were evaluated using accuracy, precision, recall, F1-score, and AUC. Because fraudulent transactions represent less than one percent of the dataset, low precision is expected across all models. In highly imbalanced settings, models can achieve high recall by flagging many transactions as potentially fraudulent, but doing so reduces precision by increasing the number of false positives. Improving precision generally requires predicting fraud less often, which reduces recall and risks missing true fraud cases. This trade-off is important to fraud detection.

Model	Accuracy	Precision	Recall	F1	AUC
Logistic Regression	0.8753	0.0248	0.8180	0.0482	0.9346
Random Forest	0.9546	0.0701	0.8762	0.1298	0.9647

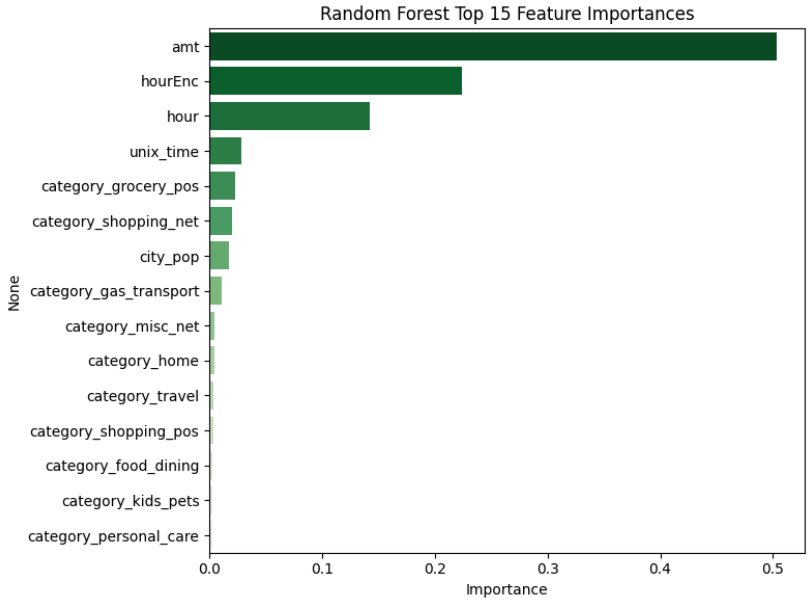
Logistic Regression achieved an accuracy of 0.8753, a precision of 0.0248, and a recall of 0.8180, resulting in an F1-score of 0.0482 and an AUC of 0.9346. These results indicate that the model successfully captured a large proportion of fraud cases but also produced many false alerts. Its advantages remain in speed, stability, and interpretability, making it a dependable baseline. The Random Forest classifier achieved stronger performance across all metrics, with an accuracy of 0.9546, a precision of 0.0701, and a recall of 0.8762, yielding an F1-score of 0.1298 and an AUC of 0.9647. The model is more effective in the separation of fraudulent and legitimate transactions. Although computationally more intensive, the Random Forest model's predictive advantages make it a strong candidate for operational deployment.



8. Model Interpretation

The Random Forest model was further examined through variable-importance analysis. Amount of transaction, transaction occurring at an abnormal hour, and hour of transaction are the top three features.

Logistic Regression is usually easy to interpret through its coefficient estimates. Although the linear model does not capture the same level of complexity as the Random Forest, the transparency of its parameter estimates adds value for risk teams seeking to explain model decisions or validate behavior against known fraud patterns.



9. Deployment Considerations

Choosing a model for deployment requires balancing predictive performance with speed. Random Forest offers stronger accuracy and more nuanced decision boundaries but may require more memory and processing time when applied to large transaction streams. Logistic Regression, while less powerful, is computationally lightweight and highly interpretable, enabling real-time scoring on millions of transactions. In practice, organizations often combine these approaches. For example, deploying Logistic Regression as a fast first-stage filter and using more complex models as a secondary review mechanism.

10. Conclusion

This project demonstrates a complete machine learning pipeline for credit card fraud detection, including data preprocessing, feature engineering, class imbalance correction, model training, and performance evaluation. Random Forest was the strongest model overall, but Logistic Regression provided an efficient and interpretable baseline, showcasing the tradeoffs between model complexity and operational demands. The results highlight the importance of thoughtful feature design, appropriate handling of imbalance, and careful model selection in addressing fraud detection challenges. Future enhancements may include incorporating more advanced algorithms, such as XGBoost.