CREDIT CARD FRAUD DETECTION

CAPSTONE PROJECT

TECHNICAL DECK

Abigail de Guzman

LICENSE & DISCLAIMER

Author: Abigail de Guzman
Course: Post Graduate Diploma in AI and Machine Learning
Institution: AIM School of Executive Education and Lifelong Learning

01

PROBLEM FRAMING

BUSINESS PROBLEM

Credit card fraud results in direct financial losses for financial institutions while also increasing operational burden due to chargebacks, investigations, and manual transaction reviews. As transaction volumes grow, relying solely on manual or rule-based detection systems becomes increasingly inefficient and costly to scale.

MACHINE LEARNING TASK

The problem is framed as a binary classification task, where each transaction is classified as either fraudulent or non-fraudulent based on historical transaction patterns. The goal of the model is to identify suspicious transactions early enough to support downstream intervention or review processes.

WHY ACCURACY ALONE FAILS

Fraudulent transactions represent less than 1% of total transactions, creating extreme class imbalance. In such settings, a model can achieve deceptively high accuracy by predicting all transactions as non-fraudulent, while still failing to detect meaningful fraud cases. As a result, alternative evaluation metrics that account for class imbalance are required.

02

DATASET

Source and Scope
The dataset used in this project is the publicly available ULB Credit Card Fraud Dataset (creditcard.csv), which contains anonymized credit card transactions made by European cardholders in September 2013.

Size and Structure
The dataset consists of 284,807 transactions with 31 variables, including transaction-level features and a binary fraud label.

Features
Transaction attributes include Time, Amount, and 28 anonymized numerical features (V1–V28) generated using Principal Component Analysis (PCA) to preserve confidentiality.

Target Variable
The target variable, Class, indicates whether a transaction is fraudulent (1) or non-fraudulent (0).

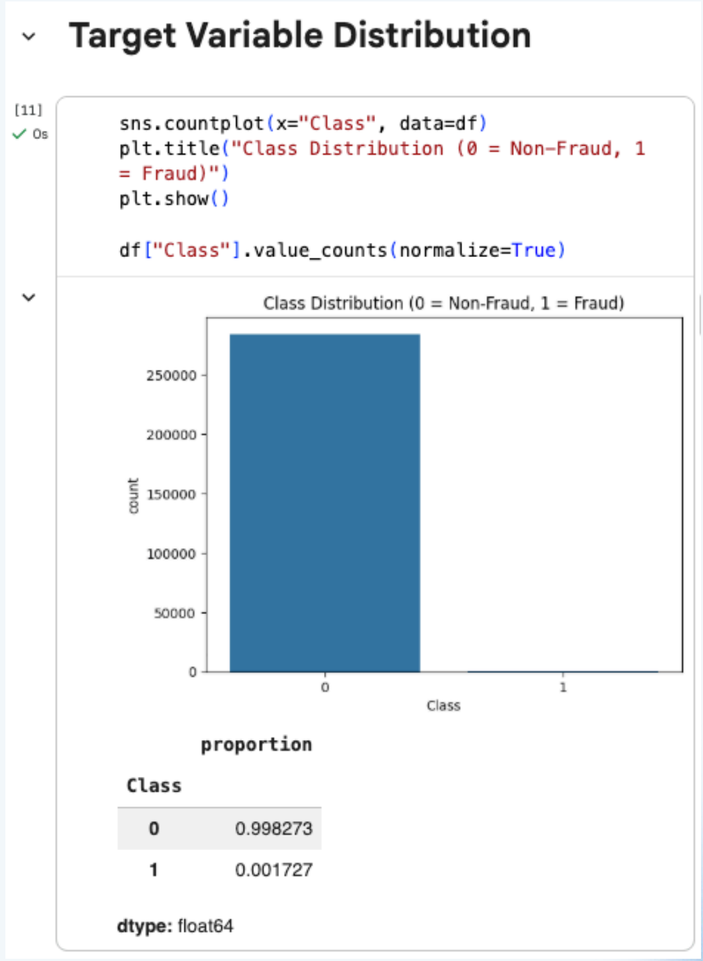
Class Imbalance
Fraudulent transactions represent less than 1% of all observations, resulting in a highly imbalanced dataset. This characteristic motivates the use of class-aware evaluation metrics and modeling techniques designed to prioritize fraud detection performance over raw accuracy.

DATA QUALITY CHECKS

The dataset was examined for missing values across all features, and no missing observations were identified. As a result, no imputation or record removal was required prior to model training.

Basic distributional checks indicate that most PCA-transformed features are centered around zero, while the original Amount feature exhibits strong right skew, reflecting the presence of high-value transactions.

The target variable shows extreme class imbalance, with fraudulent transactions accounting for approximately 0.17% of all observations. This characteristic has important implications for model evaluation and metric selection.



Class distribution showing extreme fraud imbalance in the target variable

04

PREPROCESSING

DATA PREPROCESSING PIPELINE

01

Train/test split: stratified 80/20

The dataset was split into training and test sets using a stratified 80/20 split to preserve the class distribution in both subsets.

02

Scaling: StandardScaler for LR

Feature scaling was applied using standardization prior to training the Logistic Regression model to ensure variables with different magnitudes contributed proportionally during optimization.

03

Handling imbalance

Class imbalance was addressed using class-weighted learning, assigning higher penalties to misclassification of fraudulent transactions during model training.

MODELS

MODEL SELECTION RATIONALE

Two classification models were evaluated in this project: **Logistic Regression** as a baseline and **Random Forest** as a non-linear candidate model.
These models were selected to balance interpretability, performance, and suitability for highly imbalanced fraud detection data.

BASELINE: LOGISTIC REGRESSION

Logistic Regression was used as a baseline model due to its simplicity, interpretability, and widespread use in binary classification tasks. It provides a strong reference point for evaluating more complex models and offers transparency into feature contributions, which is important in risk-sensitive domains such as financial fraud detection.

CANDIDATE: RANDOM FOREST

Random Forest was selected as a candidate model approach to capture non-linear interactions and complex decision boundaries that may not be well represented by a linear model. Model complexity was deliberately constrained through hyperparameter selection to reduce overfitting while improving fraud detection performance relative to the baseline.

MODEL 1: LOGISTIC REGRESSION

Logistic Regression was trained as a baseline binary classifier using class-weighted learning to address severe class imbalance. Feature standardization was applied prior to training to ensure stable optimization.

EVALUATION SUMMARY

ROC–AUC: ~0.97
Recall (Fraud): High (captures most fraudulent cases)
Precision (Fraud): Low (many false positives)

The model demonstrates strong discriminative ability but produces a high number of false positives when identifying fraudulent transactions.

INTERPRETATION

While Logistic Regression effectively separates fraudulent and non-fraudulent transactions overall, its linear decision boundary limits precision under extreme class imbalance. This makes it suitable as a reference baseline, but insufficient as a standalone production model.

```
lr = LogisticRegression(
    max_iter=1000,
    class_weight="balanced"
)

lr.fit(X_train_scaled, y_train)

y_pred_lr = lr.predict(X_test_scaled)
y_proba_lr = lr.predict_proba(X_test_scaled)[:, 1]

print(classification_report(y_test, y_pred_lr))
print("ROC AUC:", roc_auc_score(y_test, y_proba_lr))
```

```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99     56864
           1       0.06      0.92      0.11        98

    accuracy                           0.98     56962
   macro avg       0.53      0.95      0.55     56962
weighted avg       1.00      0.98      0.99     56962

ROC AUC: 0.9720834996210077
```

MODEL 2: RANDOM FOREST

Random Forest was trained as a non-linear ensemble classifier to improve fraud detection performance beyond linear decision boundaries. To control model complexity and reduce overfitting, tree depth and minimum leaf size were constrained, while class-weighted learning was applied to address severe class imbalance.

EVALUATION SUMMARY

ROC–AUC: ~0.976 (strong overall discrimination)
Recall (Fraud): High
Precision (Fraud): Substantially improved
False Positives: Reduced relative to baseline

The model demonstrates strong discriminative ability while substantially reducing false positives compared to a linear baseline.

INTERPRETATION

By aggregating multiple decision trees, Random Forest captures complex, non-linear interactions between transaction features. This results in improved precision for the fraud class while maintaining high recall, making the model more suitable for operational fraud detection settings.

```python
rf = RandomForestClassifier(
    n_estimators=30,         # ↓ from 100
    max_depth=10,            # caps tree complexity
    min_samples_leaf=50,     # prevents overfitting
    class_weight="balanced",
    random_state=42,
    n_jobs=-1
)

rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)
y_proba_rf = rf.predict_proba(X_test)[:, 1]

print(classification_report(y_test, y_pred_rf))
print("ROC AUC:", roc_auc_score(y_test, y_proba_rf))
```

```
...              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.62      0.87      0.72        98

    accuracy                           1.00     56962
   macro avg       0.81      0.93      0.86     56962
weighted avg       1.00      1.00      1.00     56962

ROC AUC: 0.9764099699390165
```

MODEL COMPARISON & FINAL SELECTION

08

MODEL PERFORMANCE SUMMARY

Both models achieve strong overall discrimination as measured by ROC–AUC. However, key differences emerge when considering fraud-specific performance.

Confusion matrix

KEY FINDINGS

- Logistic Regression provides a strong, interpretable baseline but suffers from low precision for the fraud class.
- Random Forest achieves higher ROC–AUC and substantially improves fraud precision while maintaining strong recall.
- This results in fewer false positives, reducing unnecessary manual reviews.

```
results = pd.DataFrame({
    "Model": ["Logistic Regression", "Random Forest"],
    "ROC AUC": [
        roc_auc_score(y_test, y_proba_lr),
        roc_auc_score(y_test, y_proba_rf)
    ]
})

results
```

	Model	ROC AUC
0	Logistic Regression	0.972083
1	Random Forest	0.976410

FINAL MODEL SELECTION

Based on the evaluation results, Random Forest is selected as the preferred model due to its superior balance between fraud detection effectiveness and operational cost.

This model provides the most practical balance between fraud detection effectiveness and operational efficiency for real-world deployment.

MODEL EXPLAINABILITY

WHY EXPLAINABILITY MATTERS

In financial fraud detection, model decisions must be interpretable to support trust, regulatory compliance, and operational review.

SHAP OVERVIEW

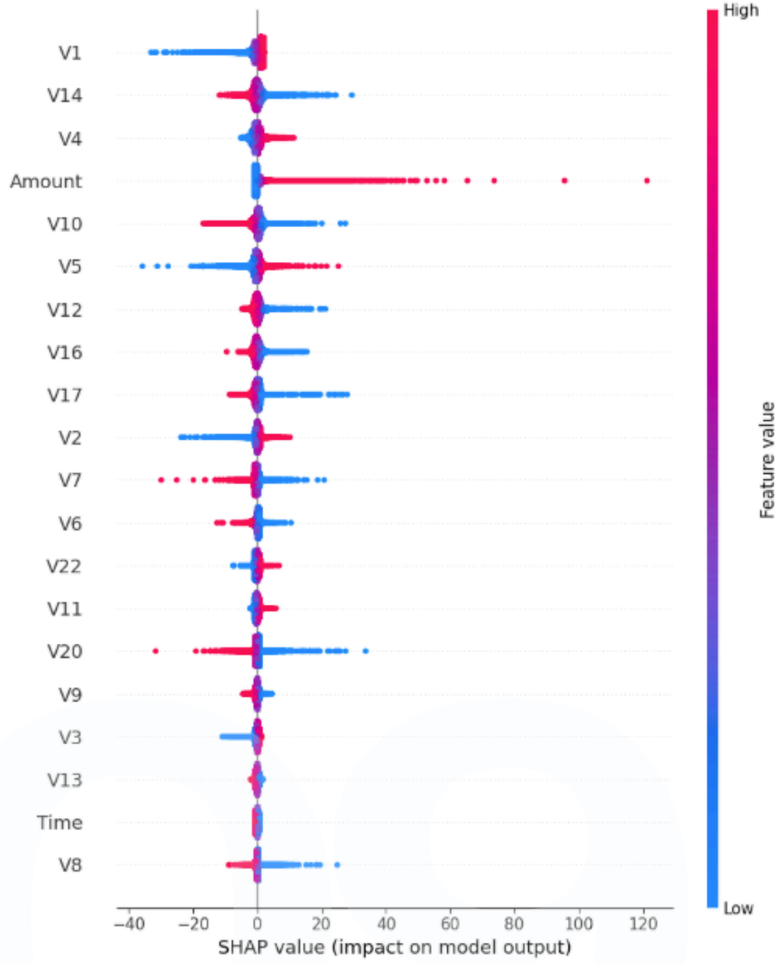
SHAP (SHapley Additive exPlanations) was used to interpret model predictions by quantifying each feature's contribution to fraud classification.

KEY INSIGHTS

- A small subset of features drives most fraud predictions
- Feature impacts are consistent across transactions
- The model does not rely solely on transaction amount, reducing bias risk

SHAP summary plot

```
shap.summary_plot(
    shap_values,
    X_test_scaled,
    feature_names=X.columns,
    show=True
)
```



10

LIMITATIONS

AND NEXT STEPS

LIMITATIONS

- Dataset reflects historical European transactions and may not generalize globally
- PCA anonymization limits semantic feature interpretation
- Threshold selection not optimized for cost-sensitive deployment

FUTURE WORK

- Cost-based threshold tuning
- Model calibration and stability testing
- Real-time deployment evaluation
- Fairness analysis across proxy variables

Acknowledgment

This deck was prepared with the aid of writing and productivity tools to support clarity and formatting:

- Grammarly was used for grammar and phrasing suggestions.
- Microsoft Copilot was used to help organize content, verify definitions, and ensure proper attribution.

In addition, concepts and examples were referenced from instructional materials provided by AIM School of Executive Education and Lifelong Learning

All final interpretations, coding steps, and written analyses were reviewed and edited to reflect the author's own understanding and insights.

END OF DECK

abbydeguzman@gmail.com